ÑO III - NÚM. 62

MICROHOBBY

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

SEMANAL

AÑO III - N.º 62

135 PTS.
Incluido IVA

CODIGO MAQUINA

INSTRUCCIONES DE CAMBIO DE SECUENCIAS

NUEVO

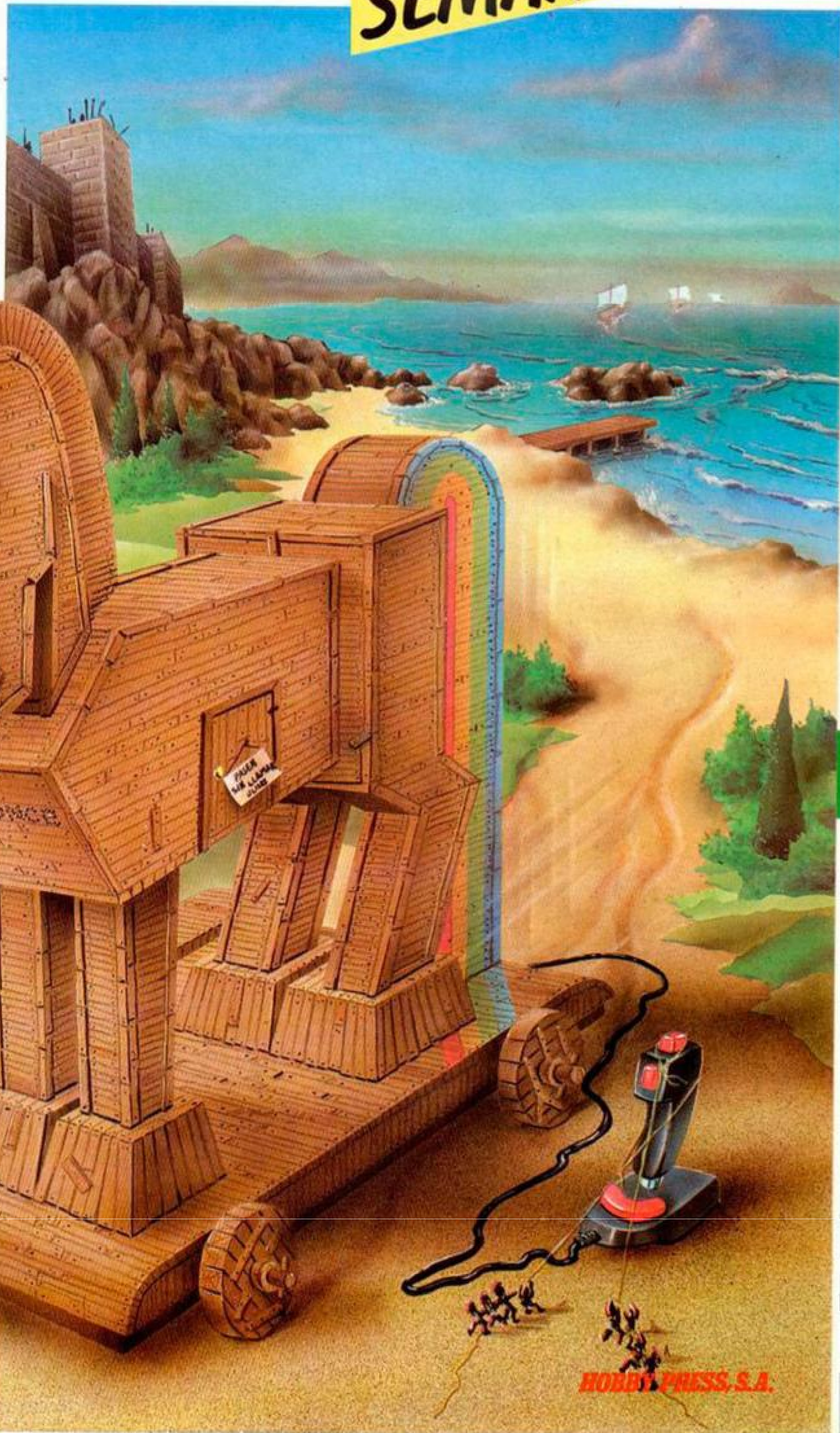
"ROBIN OF THE WOOD" EL ARQUERO DEL BOSQUE

RUTINAS DE ANIMACION

CÓMO MANEJAR LA PANTALLA DESDE C/M

MICROFILE

ALMACENAMIENTO DE INFORMACIÓN EN CLAVE



MICROHOBBY ESTA SEMANA

AÑO III. N.º 62. 21 al 27 de enero de 1986
135 ptas. (Incluido IVA)





Un entrañable héroe vuelve a nuestras páginas, «Robin of the wood». Pág. 12.

Director Editorial
José I. Gómez-Centurión

Director Ejecutivo
Domingo Gómez

Asesor Editorial
Gabriel Nieto

Redactor Jefe
Africa Pérez Tolosa

Diseño
Rosa María Capitel

Redacción
Amalio Gómez, Pedro Pérez, Jesús Alonso

Secretaria Redacción
Carmen Santamaría

Colaboradores
Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda Sergio Martínez y J. M. Lazo

Corresponsal en Londres
Alan Heap

Fotografía
Javier Martínez, Carlos Candel

Portada
José María Ponce

Dibujos
J. R. Ballesteros, A. Perera, F. L. Frontán, Pejo, J. M. López Moreno, J. Igual, J. A. Calvo, Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.

Presidente
María Andrino

Consejero Delegado
José I. Gómez-Centurión

Jefe de Publicidad
Marisa Esteban

Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76

Secretaría de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, 39
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
Rotedic, S. A. Ctra. de Irún, km. 12,450 (MADRID)

Fotocomposición
Espacio y Punto, S. A.
Paseo de la Castellana, 268

Fotomecánica
Grof
Ezequiel Solana, 16

Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cia. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

Solicitado control OJD

PREMIADOS HOBBY-SUERTE

JUAN TAMARGO GUTIERREZ. Sta. Laura, 1, 7.º D. Móstoles (MADRID).
Cinta de programas (5.ª Cat.)

JORGE VENERO ALONSO. Doce de Octubre, 6 (MADRID).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

MIGUEL CABALLERO GARCIA. Maestro Nicolau, 14. BARCELONA.
Cinta de programas (5.ª Cat.)

JOAQUIN LA CRUZ. Avda. Roma, 127, 1.º Rexonch-Lisboa (PORTUGAL).
Impresora Seikosha (2.ª Cat.)

JOSE M.ª CORONADO TORDESILLAS. Juan II, 4, 3.º B (CIUDAD REAL).
Cinta de programas (5.ª Cat.)

ANA BELEN FELTRAN ESPEJO. Florida Blanca, 12, 3.º D. Dolores (CARTAGENA)
Impresora Seikosha (2.ª Cat.)

VICTOR LEON LAGOA. Pza. de España, 18, 2.º D. Zafra (BADAJOZ).
Cinta de programas (5.ª Cat.)

MANUEL ADRADOS ANTORALZ. Castrillo de Aza, 13 (MADRID).
Cinta de programas (5.ª Cat.)

ANTONIO NAVARRO DEL PINO. Avda. del Mediterráneo, 21, 5.º A (ALMERIA).
Cinta de programas (5.ª Cat.)

DANIEL GARCIA DURAN. Avda. Ntra. Sra. Valvanera, 104. (MADRID).

ALBERTO ACUÑA AYOZA. Archiduque Luis Salvador, 90, 2.º, 5.ª (PALMA DE MALLORCA).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

JOSE ANGEL AGUILAR PEREZ. Fray Antonio de Marchena, 38. Linares (JAEN).
Cinta de programas (5.ª Cat.)

MARIANO GARICA-SALAMANCA PALAU. Marroquina, 24, 6.º D. (MADRID).
Cinta de programas (5.ª Cat.)

ELOY SERRANO BENITRAGO. Cuesta de Hista/Nuevo Alamín, 7, 5.º A (GUADALAJARA).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

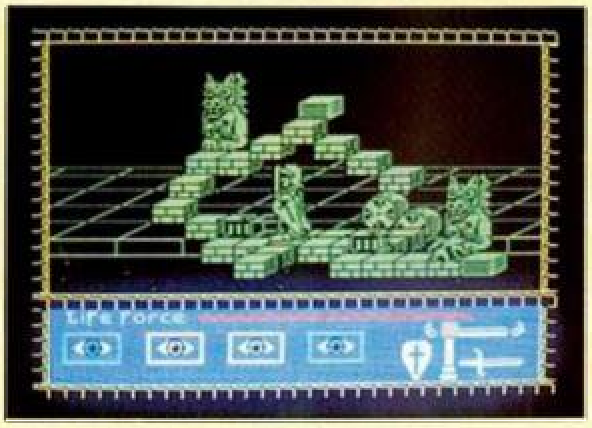
MIGUEL ANGEL ALONSO DEL PINO. Fernando Caleadillo, 19 (BADAJOZ).
Cinta de programas (5.ª Cat.)

CONCEPCION SANCHEZ TORRES. Amargura, s/n. Yebra (GUADALAJARA).
Un Spectrum 48 K (1.ª Cat.)

RASPUTIN, LA FUTURA ESTRELLA DE FIREBIRD

Firebird, la casa de software que recientemente ha alcanzado la fama con éxitos como Gyron o Elite, acaba de lanzar su última creación que, según la propia compañía y algunos entendidos en el tema, está llamada a ser una de las más destacadas estrellas de este recién estrenado año.

Se trata de Rasputín, un programa protagonizado por el mítico personaje de la Rusia de los zares, y que nos propone la misión de destruir el Diamante de los Siete Planetas, la fuente del poder del espíritu de Rasputín. Para conseguir este cometido, necesitaremos neutralizar los ocho conjuros que han sido lanzados en su nombre, utilizando al máximo nuestra habilidad y rapidez de reflejos con el fin de ir atravesando los complicados laberintos que configuran las numerosas pantallas del juego.



A modo de información y sin ningún espíritu de crítica, os presentamos una de sus pantallas para que podáis ir comprobando por vosotros mismos qué es lo que puede dar de sí este nuevo y prometedor Rasputín.

Y cambiando de tema, nos acaba de llegar una carta de la dirección de Elite que nos comunica que debido a una serie de pequeños problemas técnicos, el programa «Scooby Doo» será retrasado por un tiempo indefinido.

NUEVOS ACCESORIOS PANASONIC



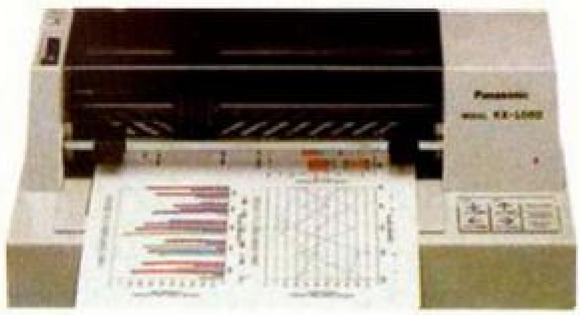
La gigantesca empresa Panasonic, multinacional con presencia en 130 países, acaba de presentar una nueva gama de productos industriales relacionados con los componentes y periféricos del ordenador.

De entre estas creaciones, que van desde modelos de ordenadores hasta los componentes e instrumentación de los mismos, destacan principalmente los siguientes:



Impresoras de gráficos: escriben mediante movimiento de una cápsula bolígrafo, posee cuatro colores con cambio normal o automático de color, hasta cuatro tipos de gráficos, varios tipos de letra y corrección y memoria para textos y gráficos.

Plotters: trazadores gráficos de tipo tambor y plano, velocidades comprendidas entre 100 y 200 mm/sg., interface Centronics o RS-232 C y formatos de papel A-3 y A-4.



Facsímil: equipos de comunicación internacional a través del teléfono, por el que se puede enviar documentos de una manera instantánea representado un ahorro del 50% sobre el coste del Télex.



Impresoras de varios tipos: matriciales, margarita, interface para conexión a distintos ordenadores, impresión en alta calidad, normal, comprimida, proporcional y gráficos.

Con MIKRO-GEN

PIERDETE EN EL PARAISO

Mikro-gen no se ha hecho mucho de rogar, y tras sus enormes éxitos obtenidos con la saga de los Week: Wally, Wilma y Herbert, se ha lanzado rápidamente de nuevo al mercado con un programa que reúne a estos tres famosos personajes en una divertida y emocionante aventura: Three Weeks in Paradise.



En esta ocasión, la familia al completo se ha ido a realizar una pequeña excursión por la selva. Pero su rudimentario transporte, un pedal, les ha dejado perdidos en pleno centro de una isla desierta.

Wilma y Herbert han sido capturados por los nativos del lugar, y Wally se ve obligado no sólo a luchar para salvar su propia vida, sino también la del resto de sus familiares.

Este juego está muy en la línea de los anteriores Everyone's a Wally o Herbert's Dummy Run tanto por los decorados y gráficos en general, como por la filosofía y desarrollo del juego, por lo que nos atrevemos a asegurar que Three Weeks in Paradise igualará o incluso superará el éxito de sus programas predecesores.

TELEFONICA Y CORNING INSTALARAN LA PRIMERA FABRICA DE FIBRA OPTICA EN ESPAÑA

En las últimas fechas se han llevado a cabo en New York las negociaciones oficiales entre Corning Glass Work y Telefónica, habiéndose llegado a un acuerdo entre ambas compañías para la creación de una empresa productora de fibra óptica en España.

La compañía Telefónica prevee alcanzar con la ayuda de Corning, el principal fabricante mundial de fibra óptica, y la empresa que posee el grado de tecnología más avanzado en este campo, un tendido de cerca de 5.000 km de cable óptico monomodo enterrado para la conexión entre las principales provincias peninsulares y otros 2.000 km de cable óptico submarino para las conexiones de Baleares y Canarias. En total se instalarán más de 90.000 km de esta fibra que unirán entre sí las ciudades más importantes del país, constituyendo la infraestructura básica de transmisión de la Red Digital Integrada (RDI).

TRUCOS



BAILE DE ATRIBUTOS

Este truco que debemos a Fernando G. de Leániz, realiza 256 pantallas diferentes, y es un programa en Código Máquina que afecta a la zona de atributos, dándole en cada pantalla un valor distinto a cada dirección de la zona de atributos.

El programa está controlado por otro en Basic, y lo que hace es pedirnos un valor para la primera dirección de memoria y acceder al programa en Código Máquina, en donde al valor inicial de la primera dirección de la zona de atributos se le irá sumando la unidad.

```
   10 FOR n=3e4 TO 30024
   20 READ a
   30 POKE n,a
   40 NEXT n
   50 INPUT n
   60 POKE 30008,n
   70 RANDOMIZE USR 3e4
   80 GO TO 50
   90 DATA 30,3,33,255,87,6,0,14,
1,22,0,35,120,119,129,71,21,194,
59,117,29,194,57,117,201
```

SONIDO Y BORDERS

Juan José Moya ha hecho una serie de trucos para hacer variaciones en los borders, y efectuar sonidos. El primero de ellos permite grabar en otro idioma, poniendo el border de color negro y la tinta también.

El segundo muestra todas las posibilidades de combinación de colores del border, papel y tinta.

El tercero consigue un border de color a la vez que emite un sonido.

Y por último, sólo desearos un buen «sonido».

```
  10 FOR f=68 TO 0 STEP -1: BEEP
.001,f
  20 OUT 254,16: BORDER 7
  30 NEXT f
```

O.K.

Si tecleamos POKE USR 4321,Ø aparecerá en la pantalla, tras pulsar el reglamentario ENTER, la sentencia Ø OK,0:1, pudiéndose escribir a continuación sin que se borre el mensaje OK. El ordenador ejecutará cualquier orden. Este truco se lo debemos a Agatha Estera.

«INMOVILIZAR PROGRAMAS»

Con este POKE 2357Ø,Ø de Fco. Javier Yáñez, conseguiremos no poder volver a ejecutar un programa mediante RUN una vez interrumpido, ni tampoco cambiar ninguna de sus líneas.

ALGER

Enrique PASCUAL

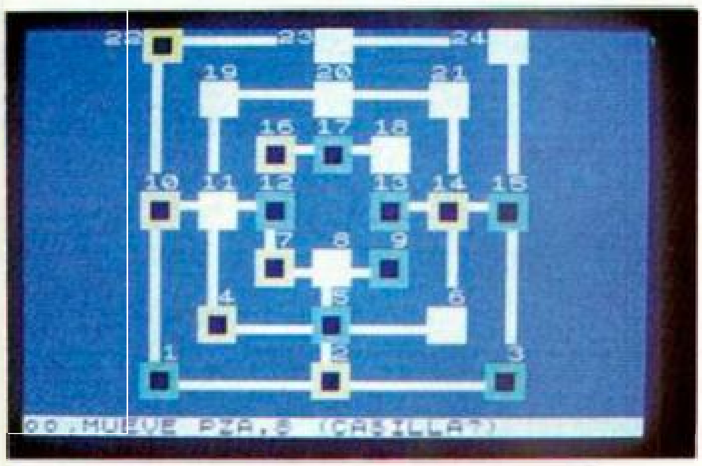
Spectrum 48 K

TODAS LAS LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO.

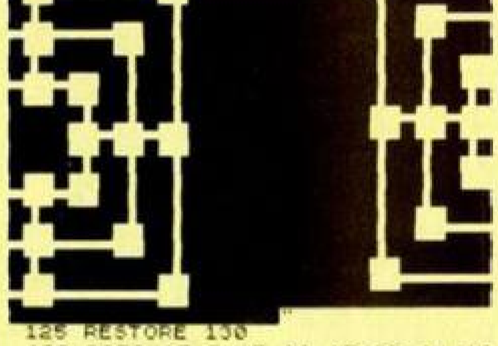
No son las tres en raya, tampoco se trata de una especialidad de damas, es simplemente ALGER, un juego de habilidad mental que te apasionará.

Se juega entre dos personas y en un tablero de 24 casillas numeradas con 9 piezas para cada jugador. Para comenzar, cada uno de ellos irá colocando sus piezas en la casilla que deseen, tratando siempre de hacer una línea vertical u horizontal o bien, tratando de encerrar al contrario. Los movimientos se realizan hacia casillas consecutivas, en vertical o en horizontal sin saltar la ficha del otro jugador, cosa que nos será permitida cuando consigamos hacer una línea.

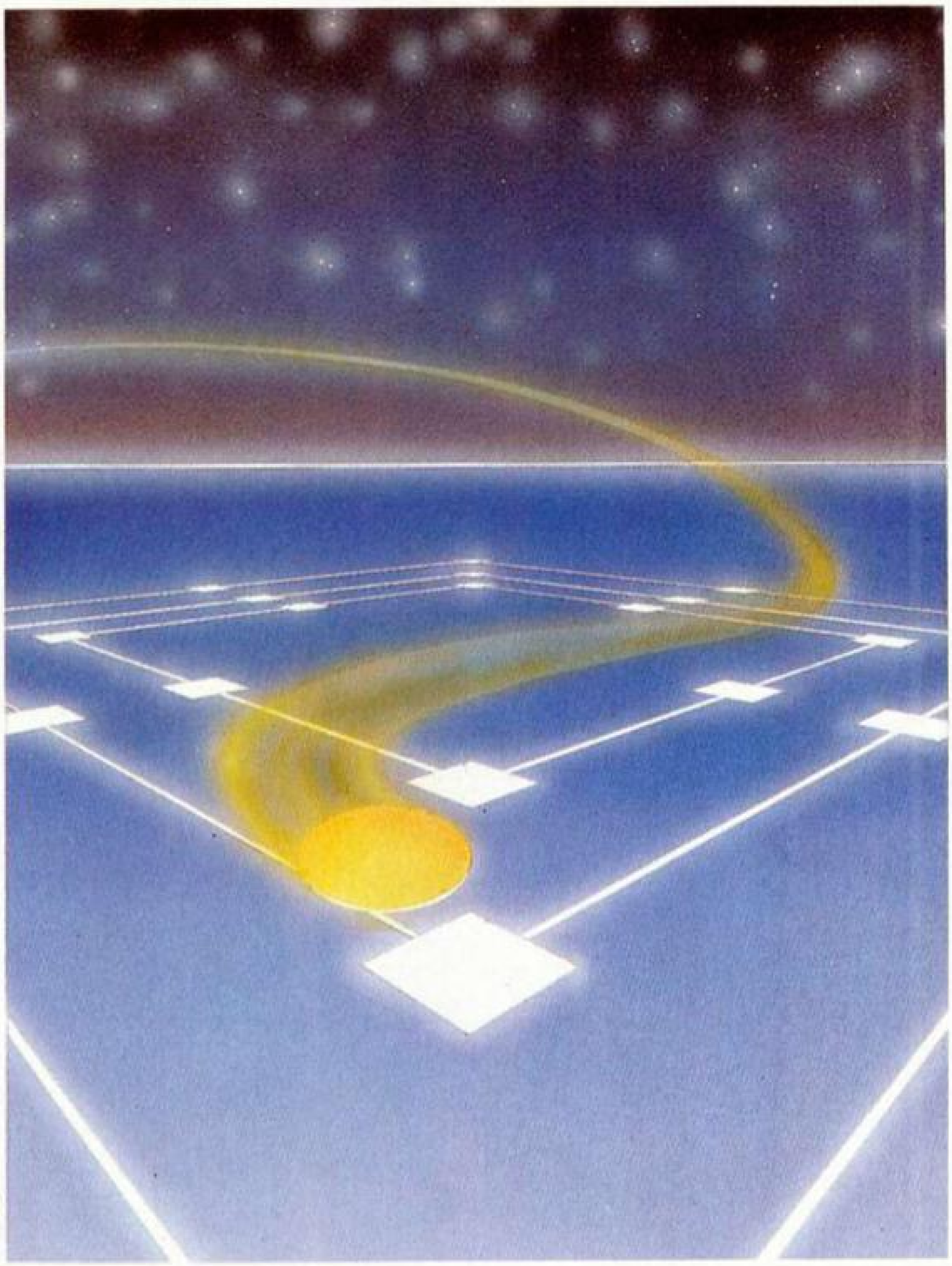
En fin, un juego de los más entretenido al que pronto cogeréis el tranquillo.




```
1 CLS
10 REM ALQUER © Enrique Pascual Lopez
20 REM VARIABLE DE ONTROL
30 DIM A(24): DIM P(24): DIM U(75)
40 FOR I=1 TO 24 STEP 1
50 LET P(I)=0: LET A(I)=0
60 NEXT I
65 RESTORE 80
70 FOR I=1 TO 75 STEP 1
80 DATA 0,0,0,6,18,19,15,18,19,24,18,19,9,15,16,15,15,16,21,15,16,12,12,13,15,12,13,18,12,13,6,9,10,9,9,10,12,9,10,18,9,10,21,9,10,24,9,10,12,6,7,15,6,7,18,6,7,9,3,4,15,3,4,21,3,4,6,0,1,15,0,1,24,0,1
90 READ U(I)
100 NEXT I
110 REM DIBUJO TABLERO
120 PRINT INK 1;"
```



```
"
125 RESTORE 130
130 DATA 17,7,17,16,17,25,14,10,14,15,14,22,11,13,11,16,11,19,8,6,8,9,8,12,8,18,8,21,8,24,5,12,5,15,5,18,2,9,2,15,2,21,0,4,0,13,0,22
140 FOR I=1 TO 24 STEP 1
150 READ T,T1
160 PRINT INK 7; PAPER 1;AT T,T1;I
170 NEXT I: BORDER 5
180 REM NOMBRE JUGADORES
190 LET I=1: GO TO 240
200 BEEP 2,5: BEEP 1,2
210 PRINT AT 21,3;"ERROR EN LOS DATOS, REPITA": PAUSE 75
220 PRINT AT 21,0;"                                "
230 IF I<>1 THEN GO TO 270
240 PRINT AT 21,0;"NOMBRE JUGADOR N.1 (8 c.m.)": INPUT N$
250 PRINT AT 21,0;"                                "
260 IF (LEN (N$)>8) OR (N$="") THEN GO TO 200
270 PRINT AT 21,0;"NOMBRE JUGADOR N.2 (8 c.m.)": INPUT X$
280 PRINT AT 21,0;"                                "
290 LET I=0: LET L=0: LET M3=1
300 IF (LEN (X$)>8) OR (X$="") THEN GO TO 200
310 REM CONTROL ENTRADA AL JUEGO
320 LET M1=0
330 LET M1=M1+1
340 LET J=1: LET O=4: LET O1=0
350 LET P$="▛▜": LET B$="▙▟"
360 LET C$=N$: LET Y$=X$
370 GO TO 410
380 LET J=2: LET O=6: LET O1=0
390 LET P$="▛▜": LET B$="▙▟"
400 LET C$=X$: LET Y$=N$
410 IF M1>9 THEN GO TO 620
420 GO TO 460
430 BEEP 2,5: BEEP 1,2
440 PRINT AT 21,3;"ERROR EN LOS DATOS, REPITA": PAUSE 75
450 PRINT AT 21,0;"                                "
460 PRINT AT 21,0;C$;",MUEVE PZA.";M1;" (CASILLA?)"
470 INPUT M
480 PRINT AT 21,0;"                                "
490 IF (M<1) OR (M>24) THEN GO TO 430
500 IF M<>INT (M) THEN GO TO 430
510 IF P(M)<>0 THEN GO TO 430
520 GO TO 1030
530 REM IMPR. PZAS. ENTRADA (M)
540 LET T=U(3*M+1)
550 LET T1=U(3*M+2)
560 LET T2=U(3*M+3)
570 PRINT INK O; PAPER O1;AT T1,T;P$
580 PRINT INK O; PAPER O1;AT T2,T;B$
590 GO TO 1260
600 IF J=1 THEN GO TO 380
610 IF J=2 THEN GO TO 330
620 REM JUGANDO EN EL TABLERO
630 PRINT AT 21,0;C$;",MUEVE DE LA CASILLA N.": INPUT M
640 PRINT AT 21,0;C$;",MUEVA A LA CASILLA N. ": INPUT M3
650 PRINT AT 21,0;"                                "
655 IF (M=0) AND (M3=0) THEN GO TO 1620
660 IF (M<1) OR (M3<1) THEN GO TO 700
670 IF (M>24) OR (M3>24) THEN GO TO 700
680 IF (M<>INT (M)) OR (M3<>INT (M3)) THEN GO TO 700
690 GO TO 740
700 BEEP 2,5: BEEP 1,2
710 PRINT AT 21,3;"ERROR EN LOS DATOS, REPITA": PAUSE 75
720 PRINT AT 21,0;"                                "
730 GO TO 630
740 IF P(M)=0 THEN GO TO 700
750 IF J<>P(M) THEN GO TO 700
760 IF P(M3)<>0 THEN GO TO 700
770 REM CONTROL PZAS(M3)SEGUN(M
780 IF M=24 THEN IF (M3=15) OR (M3=23) THEN GO TO 1030
790 IF M=23 THEN IF (M3=20) OR (M3=22) OR (M3=24) THEN GO TO 1030
800 IF M=22 THEN IF (M3=10) OR (M3=23) THEN GO TO 1030
810 IF M=21 THEN IF (M3=14) OR (M3=20) THEN GO TO 1030
820 IF M=20 THEN IF (M3=17) OR (M3=19) OR (M3=21) OR (M3=23) THEN GO TO 1030
830 IF M=19 THEN IF (M3=11) OR (M3=20) THEN GO TO 1030
840 IF M=18 THEN IF (M3=13) OR (M3=17) THEN GO TO 1030
850 IF M=17 THEN IF (M3=16) OR (M3=18) OR (M3=20) THEN GO TO 1030
860 IF M=16 THEN IF (M3=12) OR (M3=17) THEN GO TO 1030
870 IF M=15 THEN IF (M3=3) OR (M3=14) OR (M3=24) THEN GO TO 1030
880 IF M=14 THEN IF (M3=6) OR (M3=13) OR (M3=15) OR (M3=21) THEN GO TO 1030
890 IF M=13 THEN IF (M3=9) OR (M3=14) OR (M3=18) THEN GO TO 1030
900 IF M=12 THEN IF (M3=7) OR (M3=11) OR (M3=16) THEN GO TO 1030
910 IF M=11 THEN IF (M3=4) OR (M3=10) OR (M3=12) OR (M3=19) THEN GO TO 1030
920 IF M=10 THEN IF (M3=1) OR (M3=11) OR (M3=22) THEN GO TO 1030
930 IF M=9 THEN IF (M3=8) OR (M3=13) THEN GO TO 1030
940 IF M=8 THEN IF (M3=5) OR (M3=7) OR (M3=9) THEN GO TO 1030
950 IF M=7 THEN IF (M3=8) OR (M3=12) THEN GO TO 1030
960 IF M=6 THEN IF (M3=5) OR (M3=14) THEN GO TO 1030
970 IF M=5 THEN IF (M3=2) OR (M3=4) OR (M3=6) OR (M3=8) THEN GO TO 1030
980 IF M=4 THEN IF (M3=5) OR (M3=11) THEN GO TO 1030
990 IF M=3 THEN IF (M3=2) OR (M3=15) THEN GO TO 1030
1000 IF M=2 THEN IF (M3=1) OR (M3=3) OR (M3=5) THEN GO TO 1030
1010 IF M=1 THEN IF (M3=2) OR (M3=10) THEN GO TO 1030
1020 GO TO 700
1030 REM CONTR.PZA. ENTRADA (M)
1040 IF L=1 THEN GO TO 1100
1050 IF L=2 THEN GO TO 1100
1060 IF M1>9 THEN GO TO 1100
1070 LET P(M)=J
1080 LET A(M)=P(M)
1090 GO TO 530
1100 REM CONTROL PZAS (M) (M3)
1110 LET P(M)=0
1120 LET A(M)=P(M)
1130 LET T=U(3*M+1)
1140 LET T1=U(3*M+2)
1150 LET T2=U(3*M+3)
1160 PRINT AT T1,T;"  "
1170 PRINT AT T2,T;"  "
1180 IF L=1 THEN GO TO 1571
1190 LET P(M3)=J
1200 LET A(M3)=P(M3)
1210 LET T=U(3*M3+1)
1220 LET T1=U(3*M3+2)
1230 LET T2=U(3*M3+3)
1240 PRINT INK O; PAPER O1;AT T1,T;P$
1250 PRINT INK O; PAPER O1;AT T2,T;B$
1260 REM CONTROL PZAS EN LINEA
1270 IF (A(1)=A(2)) AND (A(2)=A(3)) AND (A(3)=J) THEN GO TO 1440
1280 IF (A(4)=A(5)) AND (A(5)=A(6)) AND (A(6)=J) THEN GO TO 1440
1290 IF (A(7)=A(8)) AND (A(8)=A(9)) AND (A(9)=J) THEN GO TO 1440
1300 IF (A(10)=A(11)) AND (A(11)=A(12)) AND (A(12)=J) THEN GO TO 1440
1310 IF (A(13)=A(14)) AND (A(14)=A(15)) AND (A(15)=J) THEN GO TO 1440
1320 IF (A(16)=A(17)) AND (A(17)=A(18)) AND (A(18)=J) THEN GO TO 1440
1330 IF (A(19)=A(20)) AND (A(20)=A(21)) AND (A(21)=J) THEN GO TO 1440
```



Javier Igual

```
1340 IF (A(22)=A(23)) AND (A(23)
=A(24)) AND (A(24)=J) THEN GO TO
 1440
1350 IF (A(1)=A(10)) AND (A(10)=
A(22)) AND (A(22)=J) THEN GO TO
1440
1360 IF (A(2)=A(5)) AND (A(5)=A(
8)) AND (A(8)=J) THEN GO TO 1440
1370 IF (A(3)=A(15)) AND (A(15)=
A(24)) AND (A(24)=J) THEN GO TO
1440
1380 IF (A(4)=A(11)) AND (A(11)=
A(19)) AND (A(19)=J) THEN GO TO
1440
1390 IF (A(6)=A(14)) AND (A(14)=
A(21)) AND (A(21)=J) THEN GO TO
1440
1400 IF (A(7)=A(12)) AND (A(12)=
A(16)) AND (A(16)=J) THEN GO TO

 1440
1410 IF (A(9)=A(13)) AND (A(13)=
A(18)) AND (A(18)=J) THEN GO TO
1440
1420 IF (A(17)=A(20)) AND (A(20)
=A(23)) AND (A(23)=J) THEN GO TO
 1440
1430 GO TO 1610
1440 PRINT AT 21,4;C$
1450 PRINT AT 21,13;", TIENE UNA
 LINEA": PAUSE 75
1460 PRINT AT 21,0;"
                   "
1470 PRINT AT 21,0;"PZA.A COMER
A ";Y$;",CASILLA?": INPUT M
1480 PRINT AT 21,0;"
                   "
1490 IF (M<1) OR (M>24) THEN GO
TO 1530
1500 IF (M<>INT (M)) THEN GO TO
1530
1510 IF (P(M)=0) OR (P(M)=J) THE
N GO TO 1530
1520 GO TO 1570
1530 BEEP 2,5: BEEP 1,2
1540 PRINT AT 21,3;"ERROR EN LOS
 DATOS, REPITE": PAUSE 75
1550 PRINT AT 21,0;"
                   "
1560 GO TO 1470
1570 LET L=1: GO TO 1030
1571 FOR I=1 TO 24 STEP 1
1572 IF J=1 THEN GO TO 1575
1573 IF A(I)<>1 THEN GO TO 1577
1574 GO TO 1580
1575 IF A(I)<>2 THEN GO TO 1577
1576 GO TO 1580
1577 NEXT I
1578 GO TO 1630
1580 PRINT AT 21,0;C$;", MUEVE P
ZA. DE LA LINEA": PAUSE 100
1590 PRINT AT 21,0;"
                   "
1600 LET L=2: GO TO 620

1610 LET L=0: GO TO 600
1620 LET C$=Y$
1630 PRINT AT 21,0;"HA GANADO ";
C$: PAUSE 100
1660 CLS : PRINT AT 9,4;" *** HA
STA LA PROXIMA *** ": BORDER 7
1700 REM  FIN ** ALQUER **
```



Software LO TIENE
ROS EXITOS
L 86
COSMIC WARTOAD
venturero y travieso.
sonaje que has de
smo se cuela en las
que en el vestuario
imer juego del año
COSMIC WARTOAD
En los abismos del tiempo algo sucede... Las bestias del lodo están al acecho... La guillotina galáctica desciende inexorable... La princesa está en peligro y sólo tú puedes salvarla. Las ventanas de combate te transportarán a través del tiempo y del espacio. El resto es cosa tuya.
ARE C/SANTA ENGRACIA 17, 28010 MADRID, TEL.: 447 34 10

ROBIN OF THE WOOD • Arcade • Odin

TRAS LA FLECHA DE PLATA

La leyenda sigue viva. Robin de los Bosques, el legendario héroe medieval que robaba a los ricos para repartirlo entre los pobres, vuelve a acaparar nuestra atención y nos ofrece nuevas y emocionantes aventuras.



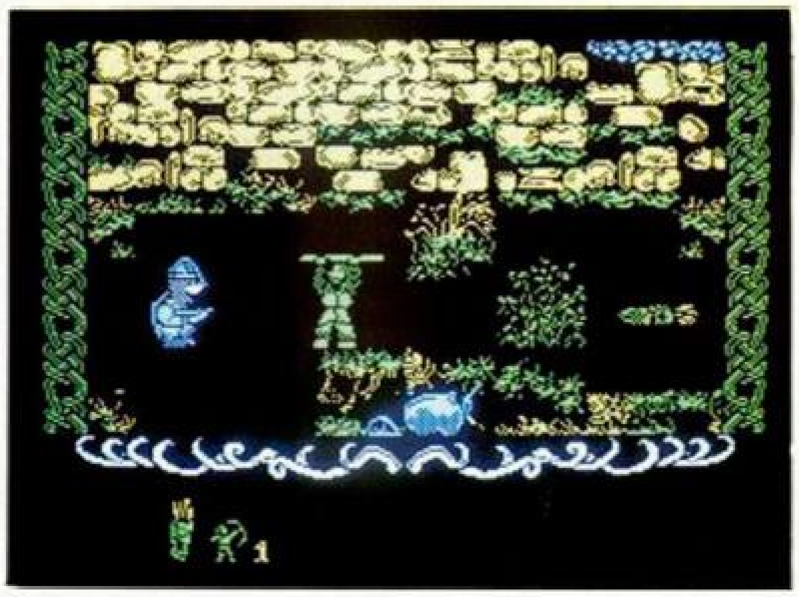
Seguro que la antigua leyenda del valeroso Robin de los Bosques ha llegado en muchas ocasiones hasta vuestros oídos. Pero posiblemente, aparte de algunos famosos episodios recordaréis muy vagamente el argumento de la historia.

Pues bien, para refrescaros un poco la memoria, os haremos un breve resumen de cómo transcurrieron los hechos de tan bella leyenda.

Todo comienza con Aleric, padre de Robin, quien era el poseedor de la gran Flecha de Plata. Esta flecha era el símbolo que garantizaba la paz y la libertad a todos los habitantes de las tierras de Saxon. Pero todo cambió tras la muerte de Aleric, pues ésta pasó a manos del Sheriff de Notingham y desde ese momento los normandos comenzaron a explotar y dominar a las gentes de Saxon.

A medida que Robin iba creciendo y se hacía cada vez más popular entre los obreros y campesinos del lugar, gracias a sus valerosas hazañas realizadas contra sus opresores, el orgullo de aquellas pobres gentes se iba haciendo mayor. Y por eso fue por lo que exigieron al Sheriff que les devolviera su tan preciada Flecha de Plata. Este se vio obligado a ceder a las presiones, por lo que decidió organizar un gran torneo. Al vencedor de dicha prueba se le entregaría como premio la Flecha de Saxon.

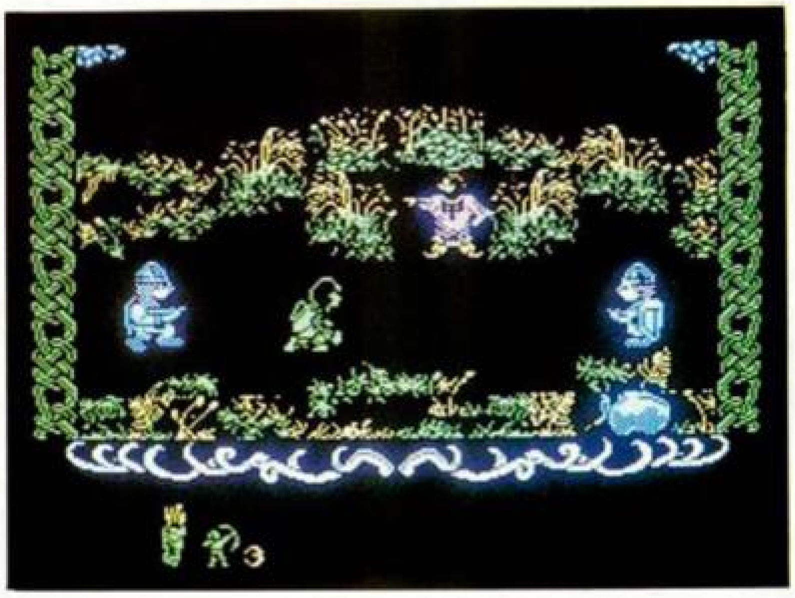
Sin embargo, todo era una trampa. El cheriff no estaba dispuesto a perder fácilmente tan valioso símbolo. Sabiendo que Robin no podría resistir la tentación de presentarse al torneo, decidió enviar al bosque a un numeroso grupo de caballeros normandos con el fin de apresarle.

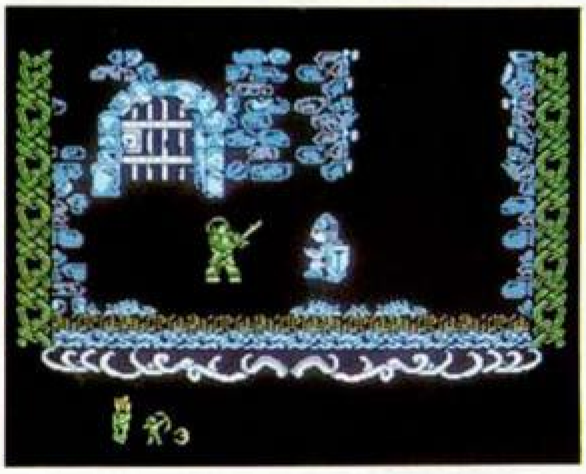




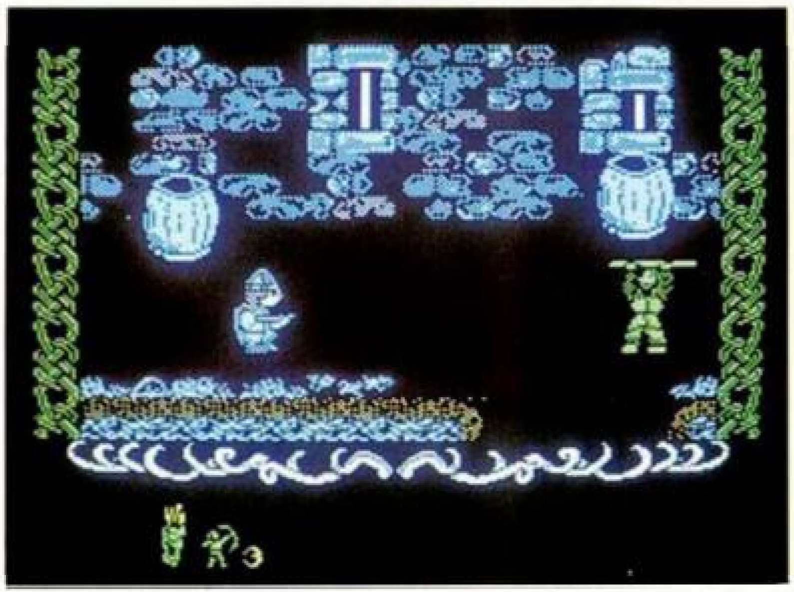
Aquí, en este preciso instante es cuando comienza el juego y en el que te toca representar el papel del acosado y perseguido Robin. Tu misión es la de recobrar la Flecha de Plata para devolver la libertad al pueblo de Saxon. Pero antes de entrar en el castillo de Notingham deberás ir realizando otra serie de cometidos. Tu arco, tu espada y tres flechas mágicas, las tiene en este momento el anciano y sabio Ent, y si quieres conseguirlas deberás entregarle tres bolsas de oro por cada una de las armas. Pero claro, para conseguir este dinero deberás robárselo al Obispo de Peterborough, quien se encuentra firmemente escoltado por un pequeño

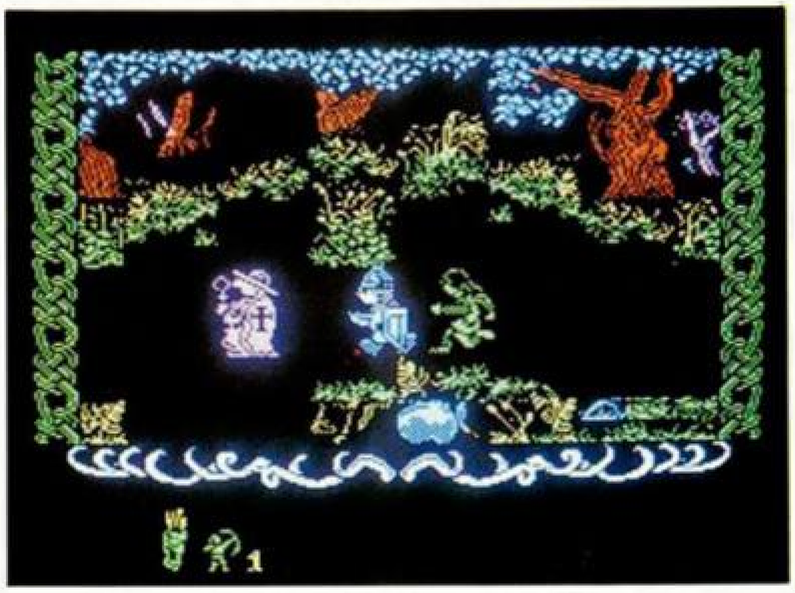








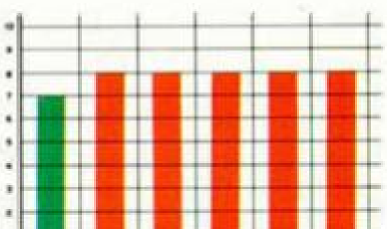




ejército de normandos, al cual deberás enfrentarte utilizando tus hábiles dotes para la pelea.

En tu largo recorrido a través del laberíntico bosque, las intrincadas calles del pueblo, los alrededores del castillo y el castillo mismo, deberás ir recogiendo algunos objetos que te irás encontrando, como flores o arcos, y también trendrás la oportunidad de ir consiguiendo algunas vidas extra, todo ello luchando contra los malvados soldados normandos, los cuales tienen órdenes explícitas de dispararte en el momento en que te vean. También deberás enfrentarte a otros enemigos que se encuentran en el interior del castillo como ciertas brujas, que aparecen y desaparecen por momentos y que te resultarán unas muy molestas adversarias.

Por todo ello, es decir, por la bella historia, las complicadas *misiones, los numerosos* peligros, los múltiples objetos y los variados escenarios, este Robin de los Bosques resulta un juego realmente entretenido, divertido y adictivo, al cual si le sumamos sus estupendos gráficos, su magnífico movimiento su bello colorido y su deliciosa sintonía y efectos sonoros, obtenemos un programa de lo más agradable e interesante de jugar.



TAU CETI • Arcade • CRL

EL PLANETA INFORTUNADO

Este Tau Ceti es un gran juego donde se dan cita la emoción, la habilidad y la visión de juego, todo ello ambientado excelentemente por unos magníficos gráficos tridimensionales y una buena presentación en pantalla.

Pero, ¿en qué consiste el desarrollo del juego?. Pues verás, tú eres el encargado de llevar a cabo la misión de salvar a éste inhospitalario y desértico planeta de una destrucción inminente.

Seguramente te estarás preguntando cómo se pudo llegar a una situación tan crítica en un planeta donde un puñado de pacíficos colonos se dedicaban a sus quehaceres galácticos cotidianos. Pues la cosa es muy simple. Desde que en el año 2050 las primeras oleadas de aventureros llegaron a Tau Ceti, todo transcurría con normalidad y las gentes veían como aquel recio planetoide iba creciendo poco a poco (en noventa años se crearon más de treinta ciudades). Sin embargo, una horrible plaga surgió de repente del frío y seco polvo de Tau Ceti y la alarma se apoderó de todos sus habitantes. La decisión se tomó rápidamente: todos fueron evacuados y regresados a la Tierra.

Pero un afamado grupo de científicos encontró sorprendentemente la cura para combatir la plaga. Sin embargo, ya era demasiado tarde. El mecanismo de seguridad de autodestrucción del planeta ya había sido activado, y el final de Tau Ceti era un proceso casi irreversible. Tan sólo con la ayuda de un experto piloto que fuera capaz de llegar hasta el reactor de fusión que alimenta al planeta se podría conseguir detener el mecanismo.

Pues sí, lo has adivinado, ese piloto eres tú. Pero recuerda que has venido hasta aquí voluntariamente, nadie te ha obligado a afrontar esta difícil y peligrosa tarea, pero ya no podrás echarte atrás. La salvación de Tau Ceti depende tan sólo de tí.

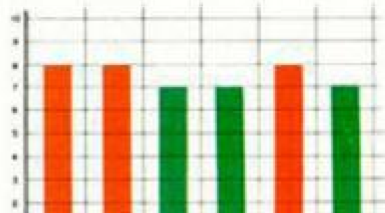






En cuanto al desarrollo del juego en sí, como podréis ver en las fotografías que os mostramos, la pantalla está dividida en varias zonas, dando cada una de ellas un tipo diferente de información. En la parte superior izquierda, tenemos un monitor que nos permite ver en todo momento el lugar en el que nos encontramos y los obstáculos y edificios que nos rodean y que, lógicamente, nos es totalmente imprescindible para poder controlar nuestros movimientos. Debajo de éste, tenemos una pequeña zona de pantalla reservada para poder comunicarnos con la computadora de nuestra nave, a través de la cual se nos permite enviar nuestras órdenes y recibir los mensajes sobre el funcionamiento de la misma. La parte derecha de la pantalla está compuesta por un radar que nos enseña los movimientos de nuestros enemigos y que es uno de los mecanismos más importantes para la consecución de nuestra misión, y por una serie de indicadores de velocidad, combustible, altura, etc., que nos ayudarán para saber en todo momento el estado de la nave.

Con todos estos aparatos a controlar, os podréis imaginar que el juego es un poco complicado y que es necesario bastante tiempo para ir conociendo las posibilidades que nos ofrece, pero con un poco de paciencia, buena voluntad, y teniendo en cuenta el precio de los juegos, seguro que acabas dándote cuenta de lo bueno que es este Tau Ceti.



SGRIZAM • Arcade • Dinamic

LA ESPADA DEL PODER





Envuelto en un gran ambiente de espectación y con aires de auténtica estrella, ha hecho su aparición en el mercado la última creación de Dinamic: Sgrizam.

Pero, sinceramente, nos ha defraudado un poco. A pesar de que los gráficos y la presentación general del programa tienen una calidad más que aceptable, el desarrollo en sí del juego resulta un tanto monótono. Los señores de Dinamic han vuelto a realizar un programa en el que, tal y como sucedía en West Bank, lo único que debemos hacer es prestar una total atención a la pantalla y mantener a prueba nuestros reflejos durante todo momento. Esto, por supuesto, no es que nos parezca mal, todo lo contrario, con juegos de este tipo se logra unos altos grados de adicción y quizás sean los que más hacen que te «piques» con ellos, pero lanzar dos programas seguidos del mismo estilo viene a ser un poco redundante.

De todas formas, y mirando el programa de una manera más objetiva, Sgrizam es un juego entretenido. La misión de éste es, (¡Oh, maravilla!), rescatar a una princesa que ha sido raptada por los malvados habitantes del planeta Kindos, quienes, además, han robado el fabuloso tesoro del Palacio Imperial. Este cometido te ha sido encomendado por el Emperador Cophenix II, quien te ha prometido como recompensa la mano de su bellísima hija, además de la mitad de sus posesiones.

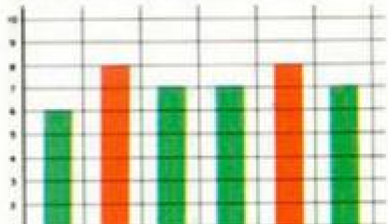
La aventura comienza en la entrada del castillo de Kindos, y deberás ir recorriendo los distintos pasadizos que conducen hasta el calabozo donde se encuentra encerrada la princesa Doxaphin. Para ello deberás ir esquivando a ciertos enemigos y enfrentándote a otros. Tienes las posibilidades de agacharte, saltar o bien desenfundar rápidamente tu espada para asestar una certera estocada a tus adversarios. Pero cuidado, todos estos movimientos deberán ser realizados con suma precisión o de lo contrario perderás una de tus tres valiosísimas vidas.



Esto es todo lo que tendrás que hacer durante el desarrollo del juego, pero cada vez necesitarás prestar una mayor atención y tener más a punto tus reflejos, pues a medida que te vayas acercando al calabozo, tus enemigos serán más peligrosos y te atacarán con mayor frecuencia.

Y ha llegado el momento de hablar de los gráficos que, como veréis, aunque tampoco son excesivamente variados, son bastante bonitos y, algo de agradecer, poseen un tamaño relativamente grande, lo que no impide que el área de la zona que se muestra en la pantalla sea también suficientemente amplia como para poder ver con claridad qué tipo de peligro se nos aproxima, y darnos tiempo para reaccionar.

En resumen, este Sgrizam es un buen juego, bien realizado y muy adictivo, pero que si no fuera por otros programas como Olé Toro o Camelot Warriors, pensaríamos que los señores de Dinamic estarían empezando a repetirse en exceso.





JR e

OBJETO:

Salta a la posición de memoria que resulta de añadir a la propia posición de la instrucción el entero de desplazamiento, el cual puede adquirir los valores desde −126 a +129.

CODIGO DE MAQUINA:

00011000	18h
<——e-2——>	

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

3

CICLOS DE RELOJ:

12

EJEMPLO:

En la dirección absoluta 45F6h

```
JR +129
```

Queremos saltar a 4677h (18039d) y estamos en 45F6h (17910) por tanto, 18039−17910=129.

Contenido del registro PC al leer la instrucción la CPU

(PC):	01000101	45h
	11111000	F8h

El registro se ha incrementado dos veces, ahora contiene 45F8h (17912d) así que el valor que hay que sumar es 7Fh (127d) es decir, "e−2".

Instrucción

JR +129:	00011000	18h
	01111111	7Fh

Observe, de nuevo, que hemos ensamblado 7Fh (127d) es decir, "e−2".

Contenido del registro PC después de la ejecución

(PC):	01000110	46h
	01110111	77h

La siguiente instrucción a ejecutar será la de la posición de memoria 4677h; observe que esta dirección es igual a 45F8h + 7Fh o, si lo prefiere, 18039 = 17912 + 127 (127= 129−2, es decir, e−2). Observe también que hemos tenido que ensamblar el valor 127 en lugar de 129, ya que el código objeto debe llevar el valor "e−2"; esto se debe a que, como decíamos antes, el desplazamiento se suma cuando el "PC" se ha incrementado dos veces para apuntar a la siguiente instrucción.

Si estuviéramos trabajando con un Ensamblador, hubiéramos puesto una etiqueta delante de la instrucción a donde queríamos saltar, por ejemplo, "LB1" y luego, habríamos hecho, simplemente:

```
JR LB1
```

El Ensamblador se hubiera encargado de calcular el desplazamiento que habría que ensamblar en el código máquina. Todas estas consideraciones en cuanto al valor del desplazamiento son igualmente válidas en todos los ejemplos que siguen sobre saltos relativos.

JR C,e

OBJETO:

Si el indicador de acarreo "C" está activo; salta a la posición de memoria que resulta de añadir a la propia posición de la instrucción el entero de desplazamiento "e", el cual puede adquirir los valores desde −126 a +129.

Si el indicador de acarreo "C" no está activo; ejecuta la siguiente instrucción.

Las instrucciones de saltos condicionales son equivalentes a la sentencia Basic:

```
IF ... THEN GO TO ...
```

CODIGO DE MAQUINA:

00111000	38h
<——e-2——>	

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

Si cumple la condición, 3

Si no cumple la condición, 2

CICLOS DE RELOJ:

Si cumple la condición 12

Si no cumple la condición 7

EJEMPLO:

En la dirección absoluta 8323h

```
JR C,+10
```


GRUPO ARITMETICO-LOGICO DE USO ESPECIAL (Indicadores y ciclos)											
	INDICADORES								No. DE	CICLOS	
NEMONICO	S	Z	x	H	x	P/V	N	C	BYTES	MEM.	REL.
DAA	♦	♦	x	♦	x	P	.	♦	1	1	4
CPL	.	.	x	1	x	.	1	.	1	1	4
NEG	♦	♦	x	♦	x	V	1	♦	2	2	8
CCF	.	.	x	x	x	.	0	♦	1	1	4
SCF	.	.	x	0	x	.	0	1	1	1	4

NOTAS:

1.- Los signos tienen el siguiente significado:

"♦": El indicador cambia de valor de acuerdo con el resultado de la instrucción.
"x": El bit adquiere un estado indeterminado.
".": El indicador conserva su anterior contenido.
"V": P/V actua como indicador de rebosamiento.
"P": P/V actua como indicador de paridad.
"0": El indicador se pone siempre a cero.
"1": El indicador se pone siempre a uno.

Fig. 6-24. Tabla resumida de indicadores y ciclos para las instrucciones aritméticas especiales.

EJERCICIOS

1.- ¿Que máscara deberíamos poner para identificar si el número, que contiene un octeto determinado, es par o impar?.

2.- Complete la siguiente rutina, de forma que barra una zona de memoria cambiando en Mayúsculas todas las letras minúsculas que se encuentre:

```
10        LD   HL,(INIC)
20        LD   BC,(LONG)
30 BUCLE  ............
40        ............
50        ............
60        INC  HL
70        DEC  BC
80        ............
90        ............
100       JR   NZ,BUCLE
110       RET
```

3.- Igual que el ejercicio anterior, pero cambiando en minúsculas todas la Mayúsculas.

4.- ¿Que resultado obtendremos en el acumulador al final de esta rutina?:

```
10        LD   A,#77
20        AND  #59
30        OR   #33
40        XOR  #15
50        RET
```

5.- Queremos entrar en una tabla "indexada" donde utilizaremos un código, contenido en el acumulador, como "ofset"; cada elemento de la tabla tiene 2 bytes de longitud; la dirección base de la tabla es 5F35h y tenemos que obtener el dato de salida en el par de registros "BC". Para ello, deberemos multiplicar por dos el contenido del acumulador (sumarlo consigo mismo) y añadirlo a la dirección base, luego, cargaremos el dato de la tabla en "BC". Complete la rutina que se encarga de hacerlo:

el indicador "P/V" está a "1" ("PE" = Parity Even, Paridad par).

Suponemos que el indicador de condiciónn "P/V" está a "1".

Contenido del registro PC al leer la instrucción la CPU

(PC):	10000101	85h
	00100011	23h

Instrucción

JP PE, #E042:	11101010	EAh
	01000010	42h
	11100000	E0h

Contenido del registro PC después de la ejecución

(PC):	11100000	E0h
	01000010	42h

La siguiente instrucción a ejecutar será la de la posición de memoria E042h. Se ha verificado el salto ya que la condición se cumplía.

EJEMPLO:

En la dirección absoluta A0A0h

JP P, #AA00

Saltar si el indicador de signo señala "positivo" ("S"=0). (JumP on Positive, Saltar si positivo).

Suponemos que el indicador de condición "S" es igual a "1".

Contenido del registro PC al leer la instrucción la CPU

(PC):	10100000	A0h
	10100011	A3h

Instrucción

	11110010	F2h
JP P, #AA00:	00000000	00h
	10101010	AAh

Contenido del registro PC después de la ejecución

(PC):	10100000	A0h
	10100011	A3h

La siguiente instrucción a ejecutar será la de la posición de memoria A0A3h. No se ha verificado el salto por no cumplirse la condición.

Instrucciones de salto relativo

JR: "Jump Relative"; salto relativo en inglés. Provocan un salto a una posición de memoria próxima a aquella desde donde se salta; la dirección del operando no viene dada en forma absoluta, sino mediante un valor de "desplazamiento" que se suma al contenido actual del "PC"; este valor puede ser positivo o negativo, lo que permite saltos hacia delante o hacia atrás y, lo que es más importante, al no figurar en la instrucción un valor absoluto, ésta funciona de igual forma independientemente de la posición de memoria donde esté colocada; dicho en otras palabras, las rutinas que utilicen saltos relativos son REUBICABLES, es decir, pueden correr en cualquier zona de la memoria.

Otra importante ventaja de los saltos relativos es que ocupan sólo 2 bytes frente a los 3 que ocupa un salto absoluto. El inconveniente fundamental es que los saltos han de ser "cortos", ya que el punto de destino debe estar muy próximo a aquel desde donde se salta. Su utilidad fundamental reside en la creación de bucles, como veremos más adelante.

Lo más engorroso de utilizar saltos relativos es calcular el valor de desplazamiento que hay que sumar al "PC" para que el salto se dirija al lugar correcto de la memoria; téngase en cuenta que el desplazamiento se suma al "PC" cuando éste ya se ha incrementado dos veces para apuntar a la siguiente instrucción. Cuando se trabaja con un Ensamblador, se suele poner una etiqueta en el punto de destino del salto, luego se hace el salto relativo a la etiqueta y el Ensamblador se encarga de la desagradable tarea de calcular el desplazamiento. No obstante, desde un principio prometimos que el curso se podría seguir sin Ensamblador, así que estudiaremos con detalle la forma de calcular saltos relativos.

Cuando se salta hacia delante no hay problema, la dificultad consiste en calcular un salto hacia atrás, ya que en este caso, el desplazamiento es negativo, es decir, complemento a 2; precisamente para esto, nos va a servir muy bien la figura 2 del capítulo relativo a los sistemas de numeración (páginas 12, 13 y 14 del curso); la segunda columna de esta figura indica el valor absoluto que se corresponde con un determinado valor negativo en complemento a 2. Veamos primero las instrucciones de salto relativo y luego volveremos sobre esto más detenidamente.

o bien, con el uso de una etiqueta:

```
ETIQUE EQU #9Ø63
       JP  ETIQUE
```

Contenido del registro "PC" inmediatamente después de que la CPU haya leído la instrucción:

(PC):	01110011	73h
	11110101	F5h

(El "PC" se ha incrementado tres veces tras leer la instrucción).

Instrucción

JP #9063:	11000011	C3h
	01100011	63h
	10010000	90h

Contenido del registro PC después de la ejecución

(PC):	10010000	90h
	01100011	63h

La siguiente instrucción a ejecutar será leída desde la posición de memoria 9063h. Se habrá producido, por tanto, un salto en el programa.

JP cc,nn

OBJETO:

Salta a la posición de memoria indicada por "nn" si la condición "cc" es verdad; en caso contrario continúa con la siguiente instrucción.

Las condiciones "cc" son las que se indican en la tabla de la figura 7-1.

CODIGO DE MAQUINA:

1 1 <—cc—> 0 1 0	
<—— n ——>	LSB
<—— n ——>	MSB

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

3

CICLOS DE RELOJ:

10

EJEMPLO:

En la dirección absoluta 8436h.

`JP NZ, LABEL`

La dirección de LABEL es 4723h.

El indicador de condición "Z" es igual a 0.

Contenido del registro PC al leer la instrucción la CPU:

(PC):	10000100	84h
	00111001	39h

Instrucción

JP NZ,LABEL:	11000010	C2h
	00100011	23h
	01000111	47h

Contenido del registro PC después de la ejecución

(PC):	01000111	47h
	00100011	23h

La siguiente instrucción a ejecutar será la de la posición de memoria 4723h, es decir, la apuntada por "LABEL".

EJEMPLO:

En la dirección absoluta 4728h.

`JP C, #4576`

Saltar a 4576h si el indicador de acarreo está a "1"; en caso contrario, continuar con la secuencia normal.

Suponemos que el indicador de condición "C" está a "0".

Contenido del registro PC al leer la instrucción la CPU

(PC):	01000111	47h
	00101011	2Bh

Instrucción

JP C, #4576:	11011010	DAh
	01110110	76h
	01000101	45h

Contenido del registro PC después de la ejecución

(PC):	01000111	47h
	00100011	2Bh

La siguiente instrucción a ejecutar será la de la posición de memoria 472Bh, es decir, no se ha producido el salto debido a que no se cumplía la condición.

EJEMPLO:

En la dirección absoluta 8520h.

`JP PE, #E042`

Saltar a la posición E042 si


```
10         LD   BC,#5F35
20         LD   H,0
30         LD   L,A
40         ADD  HL,..
50         ADD  ..,BC
60         LD   .,(HL)
70         INC  ..
80         LD   B,(..)
90         RET
```

SOLUCIONES A LOS EJERCICIOS

1.- "AND #01" o bien: "AND 1". Si el número era par, el indicador "Z" se pondrá a "1".

2.- La rutina queda:

```
10         LD   HL,(INIC)
20         LD   BC,(LONG)
30 BUCLE   LD   A,(HL)
40         AND  #DF
50         LD   (HL),A
60         INC  HL
70         DEC  BC
80         LD   A,B
90         OR   C
100        JR   NZ,BUCLE
110        RET
```

3.- En este caso, habría que cambiar la línea 40 por:

```
40         OR   #20
```

Permaneciendo, lo restante, exactamente igual.

```
4.- La operación sería la siguiente:

        Valor inicial  01110111   77h
                  AND  01011001   59h
                    =  01010001   51h
                   OR  00110011   33h
                    =  01110011   73h
                  XOR  00010101   15h
                    =  01100110   66h

    El resultado es, por tanto, 66h.

5.- La rutina quedaría:

          10      LD   BC,#5F35
          20      LD   H,0
          30      LD   L,A
          40      ADD  HL,HL
          50      ADD  HL,BC
          60      LD   C,(HL)
          70      INC  HL
          80      LD   B,(HL)
          90      RET
```

Es de suma importancia que intente comprender el funcionamiento de esta rutina, ya que se trata de un procedimiento, para moverse por tablas, que se utiliza con mucha frecuencia.

---------- o ----------

INSTRUCIONES DE CAMBIO DE SECUENCIA

Estas instrucciones son también conocidas como: «de salto».

Recordemos que la secuencia del programa se va marcando por el contenido del registro de 16 bits "PC" (Program Counter) contador de programa. La CPU lee siempre la instrucción que está en la dirección de memoria apuntada por el registro PC y añade a éste la cantidad de octetos que tiene dicha instrucción; por lo tanto el registro PC, una vez leída la instrucción por la CPU, queda apuntando a la siguiente. Si en una instrucción se cargara el registro PC con un valor distinto, la siguiente instrucción a ejecutar no sería la que le sigue secuencialmente. Pero las instrucciones que cargan el registro "PC" no existen dentro del grupo de instrucciones de carga, son de tanta importancia que se les ha reservado un grupo propio, denominado "Grupo de instrucciones de cambio de secuencia". Veámoslas una por una:

Jump, "saltar" en inglés, es el nombre genérico de estas instrucciones. A pesar de que siempre que nos referimos a ellas diremos que saltan a tal posición, lo que realmente hacen es cargar el registro "PC" con la dirección de memoria de esa posición. El utilizar el verbo "saltar" se debe a que es más descriptivo de la operación que se realiza, que cambiar de secuencia o cargar el registro "PC".

En el micro-procesado Z80 hay tres tipos de instrucciones de salto: ABSOLUTO, RELATIVO e INDIRECTO. Las de salto absoluto van a la posición de memoria marcada por el operando de la propia instrucción, bien sea mediante una etiqueta o un valor numérico. Las de salto relativo calculan la posición de memoria sumando a su propia dirección un entero de desplazamiento que puede adquirir valores comprendidos entre −126 y +129. Por último, las de salto indirecto toman la posición de memoria, a donde han de saltar, de un registro de 16 bits.

Las instrucciones de salto absoluto y relativo se pueden dividir en dos grupos: INCONDICIONALES y CONDICIONALES. Las incondicionales saltan siempre a la posición indicada; mientras que la condicionadas saltan si es cierta una condición que se les pone. Precisamente para estas últimas instrucciones se han activado los indicadores de condición en el registro "F", ahí es donde se verifica si la condición señalada es cierta o no.

Instrucciones de salto absoluto

JP: (JumP), en inglés, "salto". Simplemente, provocan un salto absoluto a la dirección de memoria indicada por el operando.

JP nn

OBJETO:

Salta a la posición de memoria indicada por "nn".

CODIGO DE MAQUINA:

11000011	C3h
<——n——>	LSB
<——n——>	MSB

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

3

CICLOS DE RELOJ:

10

EJEMPLO:

En la dirección absoluta 73F2h.

```
JP #9063
```

Observe que esta instrucción podría haber sido:

```
JP 36963
```

Profesor particular



Arturo LOBO y J. J. LEON

COMBINATORIA

Este programa está pensado para servir de ayuda a los estudiantes de 1 de BUP en una materia nueva para ellos como es la combinatoria y el binomio de Newton.

En él encontraréis cuatro posibilidades:

— Elección de la fórmula apropiada para resolver un problema, en función de las características del mismo (orden de los elementos, repetición, etc.).

— Cáculo de variaciones, combinaciones y permutaciones con o sin repetición.

— Cálculo de un elemento cualquiera de un desarrollo de Newton.

— Obtención del desarrollo de Newton completo a partir de los datos necesarios.

El programa pide en todo momento los datos que necesita. Como ejemplo de utilización podéis introducir el siguiente problema en la opción 1.

Determinar el número de jugadas distintas que nos pueden salir en el poker con baraja de 40 cartas. El número de elementos es 40, el de elementos de cada grupo es 5, no se permite repetir y no importa el orden.

TODAS LAS LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO

```
  10 LET var=1000: LET varr=1100
  20 LET com=1200: LET per=1300: LET perr=1400: LET comr=1500
 100 BORDER 5: PAPER 5: INK 3: CLS : DRAW 0,175: DRAW 255,0: DRAW 0,-175: DRAW -255,0: INK 1
 110 PRINT AT 2,13; INVERSE 1;"MENU"
 115 PRINT AT 5,0;"-Combinatoria"
 120 PRINT AT 7,4;"1)ELECCION DE FORMULA";AT 9,4;"2)CALCULOS"
 130 PRINT AT 11,0;"-Binomio de Newton";AT 13,4;"3)OBTENCION DEL TERMINO N"
 140 PRINT AT 15,4;"4)OBTENCION DEL DESARROLLO"
 200 BEEP .2,10
 210 GO SUB 1600
 215 LET y$=INKEY$
 220 IF y$="1" THEN GO TO 2000
 230 IF y$="2" THEN GO TO 3000
 240 IF y$="3" OR y$="4" THEN GO TO 280
 250 GO TO 215
 280 BEEP .1,15: BEEP .1,10
 300 CLS : PRINT INVERSE 1;"BINOMIO DE NEUTON": PRINT AT 3,0;"-Es de la forma (a+b)↑n, donde  a y b pueden ser negativos"''
 310 INPUT "Introduce:"'"a", LINE a$,"b", LINE b$,"n",n
 312 PRINT "a=";a$;" b=";b$;" n=";n'': IF y$="4" THEN GO TO 350
 315 INPUT "Que termino quieres?";m: LET m=m-1
 320 GO SUB com: PRINT "Resultado:"''TAB 4;c;"*(";a$;")↑";n-m;"*(";b$;")↑";m
 330 GO SUB 1600: GO TO 100
 350 PRINT "-El desarrollo es:"''TAB 4;: FOR m=0 TO n: GO SUB com
 360 BEEP .1,10+m: PRINT c;"*(";a$;")↑";n-m;"*(";b$;")↑";m;
 370 IF n<>m THEN PRINT " + ";
 380 NEXT m: GO SUB 1600: GO TO 100
 999 STOP
1000 LET v=1: FOR i=1 TO m
1010 LET v=v*(n-i+1): NEXT i
1020 RETURN
1100 LET vr=n↑m: RETURN
1200 LET c=1: LET l=m-(2*m-n AND 2*m-n>0): FOR i=1 TO l
1210 LET c=c*(n-i+1)/((l-i+1)+(n-i+1)*(l-i+1=0))
1220 NEXT i: RETURN
1300 LET p=1: FOR i=1 TO n
1310 LET p=p*i: NEXT i: RETURN
1400 GO SUB per: LET nd1=0
1410 FOR j=1 TO LEN m$
1420 LET nd1=nd1+(CODE m$(j)=0)
1430 NEXT j: LET nn=n: LET pp=p: LET k=0
1440 FOR j=1 TO nd1: LET m$=m$(k+1 TO ): LET k=1
1450 IF CODE m$(k)<>0 THEN LET k=k+1: GO TO 1450
1460 LET n=VAL m$( TO k-1): GO SUB per: LET pp=pp/p
1470 NEXT j: LET n=nn: RETURN
1500 LET nn=n: LET n=n+m-1
1510 GO SUB com: LET cr=c
1520 LET n=nn: RETURN
1590 PRINT AT 18,9;"(compruebalo)"
1600 PRINT #0;TAB 8;"Pulsa una tecla": PAUSE 0: RETURN
2000 BEEP .1,15: BEEP .1,10
2005 CLS : PRINT "Veamos...": INPUT "Cuantos elementos distintos hay?";n: PRINT AT 1,1;"Tengo ";n;" elementos"
2010 PRINT AT 3,1;"-Mi problema es averiguar cuan-tos grupos puedo hacer con ellos"
2020 INPUT "Cuantos elementos tiene cada   grupo? ";m
2025 PRINT AT 3,0;TAB 31;" ";TAB 31;" "
2030 PRINT AT 3,1;"Quiero formar grupos de ";m;" ele-"'" mentos cada uno"
2035 IF m>n THEN LET rep=1: LET b$="": GO TO 2050
2040 INPUT "Podemos repetir elementos? (s/n)"; LINE a$: LET rep=0: LET b$="No ": IF a$="s" OR a$="S" THEN LET rep=1: LET b$=""
2050 PRINT AT 6,1;b$;CHR$ (112-32*rep);"odemos repetir";
2055 IF m>n THEN PRINT ", claro"
2060 INPUT "Importa el orden? (s/n) "; LINE a$: LET ord=0: LET b$="No ": IF a$="s" OR a$="S" THEN LET ord=1: LET b$=""
2070 PRINT AT 8,1;b$;CHR$ (105-32*ord);"mporta el orden"
2080 PRINT AT 10,1;"Hay que aplicar la formula de:"''" ";: INVERSE 1
2090 IF NOT ord THEN LET p$="com": PRINT "Combinaciones ";: GO TO 2120
2100 IF m>=n THEN LET p$="per": PRINT "Permutaciones ";: GO TO 2120
2110 LET p$="var": PRINT "Variaciones ";
2120 IF rep THEN PRINT "con repeticion": LET p$=p$+"r"
2130 INVERSE 0: LET z=0: BEEP .2,10: GO SUB 1600: GO TO VAL p$+3000
3000 CLS : PRINT AT 2,11; INVERSE 1;" Opciones "''
3005 BEEP .1,15: BEEP .1,10
3010 PRINT " 1)VARIACIONES"''" 2)VARIACIONES CON REPETICION"''" 3)PERMUTACIONES"''" 4)PERMUTACIONES CON REPETICION"''" 5)COMBINACIONES"''" 6)COMBINACIONES CON REPETICION"
3020 GO SUB 1600
3030 IF CODE INKEY$<49 OR CODE INKEY$>54 THEN GO TO 3030
3040 LET opc=VAL INKEY$: RESTORE 3050: FOR z=1 TO 6: READ nl,p$: IF opc=z THEN GO TO nl
3045 NEXT z
3050 DATA 4000,"var",4100,"varr",4300,"per",4400,"perr",4200,"com",4500,"comr"
4000 CLS : PRINT "VARIACIONES": BEEP .1,5: PRINT AT 3,0;"V n,m=n(n-1)(n-2)...(n-m+1)"
4005 IF z THEN GO SUB 4900
4010 GO SUB 4970: PRINT v: GO SUB 1590: GO TO 100
4100 CLS : PRINT "VARIACIONES CON REPETICION": BEEP .1,5: PRINT AT 3,0;"VR n,m=n↑m"
4105 IF z THEN GO SUB 4900
4110 GO SUB 4970: PRINT vr: GO SUB 1590: GO TO 100
4200 CLS : PRINT "COMBINACIONES": BEEP .1,5: PRINT AT 3,0;"C n,m=n!/(m!(n-m)!) =V n,m/m!"
4205 IF z THEN GO SUB 4900
4210 GO SUB 4970: PRINT c: GO SUB 1590: GO TO 100
4300 CLS : PRINT "PERMUTACIONES": BEEP .1,5: PRINT AT 3,0;"P n=n!=n(n-1)(n-2)...3*2*1"
4305 IF z THEN INPUT "Introduce n ";n
4310 GO SUB 4970: PRINT p: GO SUB 1590: GO TO 100
4400 CLS : PRINT "PERMUTACIONES CON REPETICION": BEEP .1,5: PRINT AT 3,0;"PR n,(x,y...)=n!/(x!y!..)"''"(uno se repite x veces,otro y...)"
4404 IF z THEN INPUT "Introduce n ";m
4407 INPUT "Cuantos elementos se repiten? ";h: LET m$="": FOR k=1 TO h: INPUT "Introduce cuantas veces se repi-te el elemento ";(k),a: LET m$=m$+STR$ a+CHR$ 0: NEXT k
4410 LET n=m: GO SUB 4970: PRINT pp: GO SUB 1590: GO TO 100
4500 CLS : PRINT "COMBINACIONES CON REPETICION": BEEP .1,5: PRINT AT 3,0;"CR n,m =(n+m-1)!/(m!(n-1)!)"
4505 IF z THEN GO SUB 4900
4510 GO SUB 4970: PRINT cr: GO SUB 1590: GO TO 100
4900 INPUT "Cuanto vale n? ";n'"Cuanto vale m? ";m: RETURN
4970 PRINT AT 10,2;"El resultado es: ";: GO SUB VAL p$: RETURN
```



ARCHIVO DE DATOS EN CLAVE

J. M. FRAILE

Cualquiera que sea el periférico utilizado para almacenar nuestros datos y programas, es posible la aplicación de estas sencillas subrutinas para conseguir que la información no pueda ser utilizada por otras personas.

Las dos rutinas que presentamos a continuación son completamente reubicables, por lo que pueden utilizarse en cualquier posición de la memoria, siempre que no interfiramos en los propios bytes que deseamos cifrar.

La primera de ellas permite, gracias a la operación lógica XOR, cambiar los contenidos de las posiciones de memoria del bloque de direcciones que indiquemos. Esta operación lógica actúa como máscara con un determinado valor al que llamaremos CLAVE, y que puede ser fijado por nosotros. El valor de la Clave debe estar comprendido entre 1 y 255 (el cero no tendría ningún efecto apreciable).

Básicamente, la operación lógica XOR lo que hace es comparar cada bit de un determinado byte con otro contenido en el acumulador (registro A).

Por ejemplo:

byte 1	Ø1Ø111Ø1	Información
byte 2	1Ø11Ø1ØØ	Clave
XOR	111Ø1ØØ1	Cifrado

Si los dos bits comparados son distintos, el resultado será un 1, y si ambos son iguales, un Ø.

Un bloque de bytes, ya sea una pantalla, un programa en Código Máquina o un texto de procesador, puede ser fácilmente cifrado. Sin embargo, este método no puede aplicarse a un programa en BASIC, ya que al hacer la operación lógica XOR con la zona de BASIC, el programa perdería todo sentido para el intérprete y, probablemente, para el resto del sistema operativo. De esta forma sería imposible controlar al ordenador, lo que sin duda no es nuestro objetivo. Para aplicar este sistema a un programa BASIC, es imprescindible salvarlo como Bytes y luego, someter este bloque de bytes a la operación de cifrado cargándolo en otra zona de memoria que no sea la ocupada por el programa BASIC.

SENCILLO PERO UTIL

Esta pequeña rutina es, sin embargo, lo suficientemente potente como para dejar totalmente irreconocible una determinada información.

Si se dispone de un ensamblador, puede utilizarse el listado adjunto del desensamblador de la RUTINA 1, y si no, se puede teclear el código objeto directamente valiéndonos del programa BASIC correspondiente.

Al ejecutarlo nos pedirá la dirección inicial de los bytes a cifrar, el número de bytes y la CLAVE.

Instantáneamente la información quedará cifrada.

Hay que tener en cuenta que la rutina cifradora se ubica en la dirección 23296, es decir, en el buffer de impresora, por lo que si se desea cifrar el contenido de estas direcciones de memoria, habrá que buscarle otro lugar más idóneo a nuestra rutina. Para ello nada más fácil que modificar el listado Basic en consecuencia. Con unos pocos conocimientos de lenguaje máquina será fácil ubicar la rutina en la dirección de RAM deseada, sin programa BASIC, introduciendo la CLAVE mediante POKES y activándola con un RANDOMIZE USR.

El inconveniente principal de esta primera rutina es que «sólo» dispone de 255 claves distintas, por lo que con bastante paciencia puede descifrarse el Código. ¡Sólo hay que probar 255 veces!

OTRO METODO MAS SOFISTICADO

Sin embargo, aunque la segunda rutina es casi tan simple como la primera, los resultados obtenidos con su utilización son bastante más complejos de descifrar.

Básicamente consiste en hacer la misma operación lógica pero esta vez, en lugar de hacerlo siempre con un mismo valor, éste varía según la posición que ocupe en le ROM. En otras palabras, compara nuestro programa con la ROM y genera un Código Cifrado que sólo podrá ser correctamente interpretado si se conoce la posición, dentro de la memoria ROM en que se ha iniciado la comparación. Este valor será la CLAVE en este caso. De esta forma, las posibilidades aumentan hasta 16384.

Ahora bien, hay una limitación. La comparación no puede efectuarse con la RAM, ya que ésta varía sus contenidos con frecuencia, incluso de manera incontrolable para el usuario (stack, variables del sistema, etc.), lo que provocaría resultados erróneos. Así, el máximo número de bytes a cifrar no podrá ser mayor de 16384 (para no vernos obligados a comparar con la RAM), y por otra parte, la clave (dirección donde se inicia la comparación) no podrá ser nunca superior al resultado de restar el número de bytes de 16384. Es decir, si queremos cifrar 1245 bytes, la clave deberá estar comprendida entre Ø y 15139 ya que si es mayor de esta cifra, por ejemplo 15339, la comparación de los últimos 200 bytes se llevaría a cabo con la RAM de pantalla. Al tratar de descifrar la información, con bastante probabilidad la RAM de pantalla será distinta, y el resultado obtenido no tendrá nada que ver con el original.

DESCIFRAR

Por supuesto, cualquier información Cifrada no podrá utilizarse correctamente hasta que no haya sido convenientemente Desci-

frada. Para ello no hay más que llamar de nuevo a la rutina de cifrado con el mismo valor de CLAVE. De esta forma el bloque quedaría como al principio.

En el ejemplo anterior se ve claramente:

byte 1 Cifrado	11101001	
byte 2 Clave	10110100	
XOR Descifrado	01011101	

De esta forma tan sencilla la información quedará complatamente restituida.

CLAVES ANIDADAS

Por supuesto, puede cifrarse por el segundo método un bloque que haya sido cifrado ya por el primero, y viceversa. De esta forma se consigue complicar todavía más la información.

La única norma que hay que respetar escrupulosamente, es que al descifrar las claves se apliquen en orden inverso a como se utilizaron al Cifrar.

Por ejemplo. Si se ha aplicado la CLAVE 14 con la primera rutina y la CLAVE 1234 con la segunda, para proceder al descifrado hay que utilizar primero la segunda rutina con la CLAVE 1234 y luego la primera con la CLAVE 14.

Lo que no tiene mucho sentido, en todo caso, es tratar de utilizar el primer método repetidas veces sobre un mismo bloque de información, aunque sea con distintas claves, ya que el resultado obtenido, por muy complejas manipulaciones que hayamos realizado, podrá siempre descifrarse con una sola clave.

Veamos un ejemplo:

01011101 Información original

10100110 1.er cifrado (clave 166)

11111011 resultado

10101010 2.º cifrado (clave 170)

01010001 resultado

11100111 3.er cifrado (clave 231)

10110110 resultado

11101010 descifrado (clave 234)

01011101 Información original

DESENSAMBLE DE LA RUTINA 1

```
10        LD   DE,NUMERO DE BYTES
20        LD   HL,DIRECCION DE INICIO
30 CIFRAR LD   A,(HL)
40        XOR  CLAVE
50        LD   (HL),A
60        INC  HL
70        DEC  DE
80        LD   A,D
90        OR   E
100       JR   NZ,CIFRAR
110       RET
```

PROGRAMA 1

```
10 FOR N=23296 TO 23313
20 READ A:POKE N,A:NEXT N
30 DATA 17,0,0,33,0,0,126,238,0,
119,35,27,122,179,32,246,201
40 INPUT "DIRECCION INICIAL: ";D
IR
50 INPUT "NUMERO DE BYTES: ";NUM
60 INPUT "CLAVE: ";CLA
70 POKE 23298,INT(NUM/256):POKE
23297,(NUM-256*(INT(NUM/256)))
80 POKE 23301,INT(DIR/256):POKE
23300,(DIR-256*(INT(DIR/256)))
90 POKE 23304,CLA
100 RANDOMIZE USR 23296
```

DESENSAMBLE DE LA RUTINA 2

```
10        LD   DE,NUMERO DE BYTES
20        LD   BC,DIRECCION DE INICIO
30        LD   HL,CLAVE
40 CIFRAR LD   A,(BC)
50        XOR  (HL)
60        LD   (BC),A
70        INC  BC
80        INC  HL
90        DEC  DE
100       LD   A,D
110       OR   E
120       JR   NZ,CIFRAR
130       RET
```

PROGRAMA 2

```
10 FOR N=23296 TO 23315
20 READ A:POKE N,A:NEXT N
30 DATA 17,0,0,1,0,0,33,0,0,10,1
74,2,3,35,27,122,179,32,246,201
40 INPUT "DIRECCION INICIAL: ";D
IR
50 INPUT "NUMERO DE BYTES: ";NUM
60 INPUT "CLAVE: ";CLA
70 POKE 23298,INT(NUM/256):POKE
23297,(NUM-256*(INT(NUM/256)))
80 POKE 23301,INT(DIR/256):POKE
23300,(DIR-256*(INT(DIR/256)))
90 POKE 23304,INT(CLA/256):POKE
23303,(CLA-256*(INT(CLA/256)))
100 RANDOMIZE USR 23296
```

COMO MANEJAR LA PANTALLA DESDE CODIGO MAQUINA

Paco PORTALO

Una de las tareas más usuales en programación de juegos es el tratamiento de la pantalla. En este artículo nos introduciremos en la forma de manejarla de una manera efectiva, con el fin de sacarle el máximo partido.

Hay tres tipos de programadores a los que puede resultar bastante útil la rutina que veremos a continuación, y diferenciaremos para cada caso la forma de mayor aprovechamiento.

En primer lugar, aquellos que desean introducirse por las sendas del código máquina para sacar todo el partido posible a su Spectrum, a estos daremos algunas claves fundamentales a la hora de programar en el «temible» lenguaje ensamblador (no tanto).

Por otra parte, los ya avezados en código máquina seguro que encontrarán algunos trucos que les serán de gran utilidad, a estos les proponemos un juego entretenido y de gran interés si lo suyo es el código máquina, intentando descubrir el porqué de las rutinas que veremos a continuación, sin haber visto antes las explicaciones, claro está.

Por último, tenemos a aquellos que lo único que les interesa es una rutina en máquina que les haga tal o cual cosa, sin importarles el cómo ni el por qué, es decir, hacen sus pokes desde un programa Basic, hacen una llamada con alguna de las sentencias USR, y quieren obtener un resultado. Cada uno conseguirá lo que quiere.

Veremos una rutina fundamental y bastante depurada, que tiene por objeto coordenar la pantalla del Spectrum pixel a pixel (o por caracteres), de una forma rápida y elegante.

Sus aplicaciones pueden ser múltiples y variadas, como por ejemplo, hacer un plot en un punto determinado de la pantalla, representar una función matemática o saber exactamente qué dirección de pantalla corresponde a unas determinadas coordenadas (fundamental si queremos hacer mover una figura a lo largo de la pantalla). La llamaremos Rutina CO (coordenación).

Para entender esta rutina, es necesario conocer perfectamente la arquitectura de la pantalla del Spectrum, esto es, como están los bytes organizados, y aunque ya se ha hablado mucho de esto, no está de más un ligero repaso.

Estructura de la pantalla

La pantalla del Spectrum está dividida en tres tercios, cada tercio contiene 32 columnas y 8 filas de caracteres, y si un carácter tiene 8 bytes, esto quiere decir que un tercio tiene 8x32x8=2048 bytes, y si multiplicamos por 3 tercios que tiene la pantalla, tenemos 6144 bytes (6 kbytes).

Lo complicado de esta arquitectura es que cada tercio es independiente, quiere decir esto que, si pokeamos desde la dirección donde empieza la pantalla hasta su final, primero se nos llenará el primer tercio, después el segundo y por último el tercero, además observaremos que primero se nos llena la primera línea de la segunda fila de caracteres, después la primera línea de la segunda fila de caracteres, y así sucesivamente hasta llegar a la octava fila; después empezaremos por la segunda línea de la primera fila, etc...

Para entender esto mejor, hagamos el siguiente programa Basic:

```
10 FOR A=16384 TO 22527: PAUSE
0: POKE A,225: NEXT A
```

Al presionar una tecla, el programa avanza. Puede detenerse, soltando, para una mejor comprensión.

Como seguro que casi todos sabréis, la pantalla en el Spectrum está en una zona real de memoria que empieza en la dirección 16384=#4000 (utilizaremos la almohadilla siempre que nos refiramos a un número en hexadecimal).

Coordenación de caracteres

Una vez explicada la pantalla, veremos la rutina de coordenación para aquellos que deseen saber de que va sin más explicaciones. Animo, que seguro será un ejercicio de gran utilidad para perfeccionar vuestros conocimientos del lenguaje en código máquina. Como referencia, sabed que en el registro C irá la columna, y en el B la fila, dándonos el resultado en el registro par HL. Por ahora sólo coordenaremos por caracteres (no por pixels), es decir, que los valores máximos que pueden tomar los registros B y C son 23 y 31 respectivamente.



```
10   LD   H,#40
20   LD   A,B
30   AND  24
40   ADD  A,H
50   LD   H,A
60   LD   A,B
70   AND  7
80   RRCA
90   RRCA
100  RRCA
110  ADD  A,C
120  LD   L,A
130  RET
```

Esta rutina, a pesar de su brevedad, es bastante potente y efectiva. Hacer programas es como trabajar escribiendo libros, se puede decir lo mismo de mil formas diferentes, y en esto está la imaginación de cada uno.

Cuando nos planteamos hacer



una rutina en código máquina, es importante conocer *qué es lo que queremos hacer* y cuál sería la forma más efectiva y rápida para desarrollarlo. En el caso que nos ocupa, lo primero será estudiar la relación que existe entre los dos bytes que utilizaremos de coordenadas (BC) y los dos bytes de resultado (en el registro doble HL), que será la dirección a que corresponden dichas coordenadas.

Para ello vamos a trabajar en Binario, pues nos será mucho más clarificador y útil. Supongamos las coordenadas Ø,Ø.

REGISTRO	B	C
BINARIO	ØØØØØØØØ	ØØØØØØØØ
HEXA.	ØØ	ØØ

Este sería el resultado: # 4ØØØ = 16384.

REGISTRO	H	L
BINARIO	Ø1ØØØØØØ	ØØØØØØØØ
HEXA.	4Ø	ØØ

Como veis, es muy simple cuando se trata de Ø,Ø. Pero imaginemos que se tratara de las coordenadas: fila 7, columna 1Ø. En este caso introduciremos los siguientes valores:

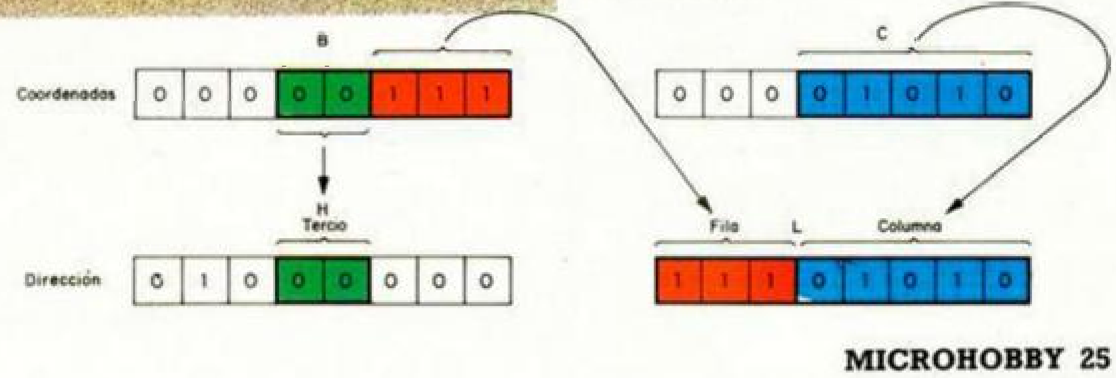
REGISTRO	B	C
BINARIO	ØØØØØ111	ØØØØ1Ø1Ø
HEXA.	Ø7	ØA

Y la dirección resultante será:

REGISTRO	H	L
BINARIO	Ø1ØØØØØØ	111Ø1Ø1Ø
HEXA.	4Ø	EA

Podríamos seguir haciendo esta operación varias veces, y llegaríamos a una conclusión muy interesante: la dirección resultante en el registro doble HL no es sino una composición de los valores expresados en los registros B y C.

Relación existente entre las coordenadas (registro BC) y la dirección de pantalla (registro HL).



Esta relación es correcta y real, si la estudiáis detenidamente descubriréis su lógica.

DESARROLLO DE LA RUTINA

Veamos cómo se puede expresar esta relación en código máquina. Primero calcularemos la fila, y a continuación la columna, puesto que ésta es más sencilla:

LD H, # 4Ø: Metemos en el registro H el valor de comienzo de pantalla. Sería lo mismo que meter en HL # 4ØØØ.

LD A,B: Introducimos en el acumulador el contenido del registro B para operar.

AND 24: Cogemos los bits 3 y 4 de la coordenada de línea. Estos determinan el tercio. En estos bits puede haber los siguientes valores:

Ø Ø - primer tercio
Ø 1 - segundo tercio
1 Ø - tercer tercio

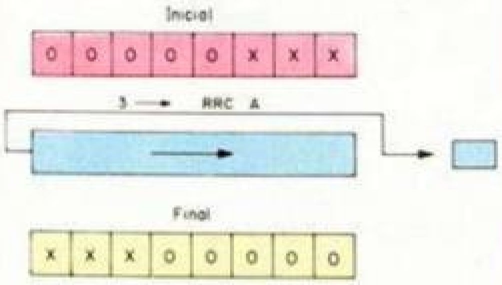
ADD A,H: Sumamos el acumulador al registro H.

LD H,A: Al estar compuesto el byte alto de la dirección, lo salvamos definitivamente en el registro H.

LD A,B: Volvemos a coger B para operar.

AND 7: Tomamos los tres bits menos significativos.

RRCA: Con esta operación de rotación trasladamos los tres bits menos significativos a la parte alta del byte. (Los tres más significativos.)



Traslado de los tres bits menos significativos a la parte alta del byte, mediante la instrucción RRCA.

ADD A,C: Sumamos el valor del registro C (donde tenemos la columna) al acumulador.

LD L,A: Salvamos el resultado en el registro L, con lo que tenemos completa la dirección en el registro doble HL.

RET: FIN.

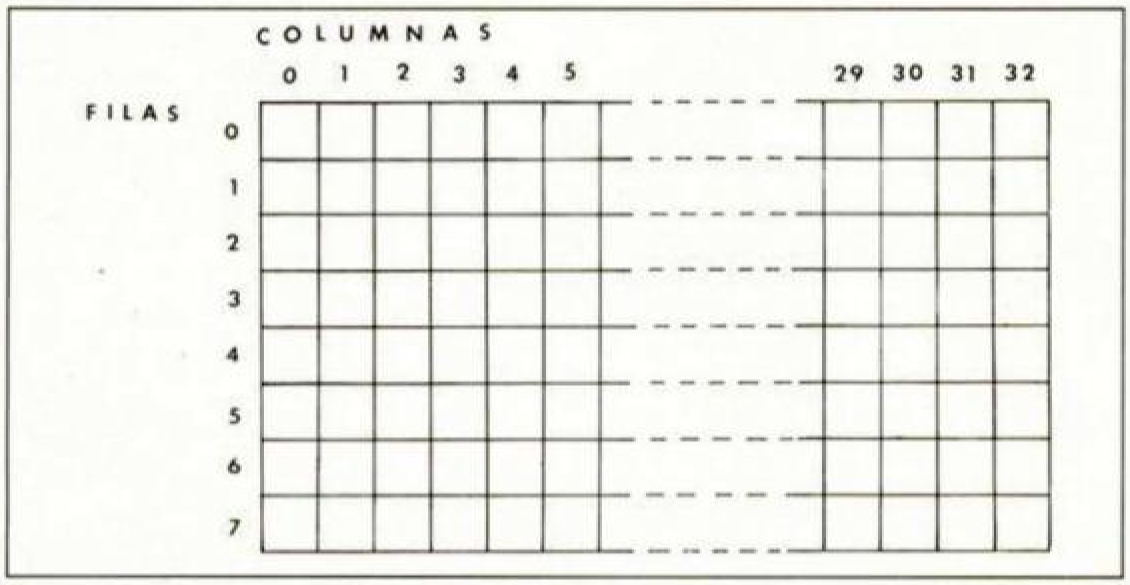
Ya tenemos en HL la dirección donde comienza el carácter dado por las coordenadas B-fila y C-columna.

Bueno, diréis que esto no hace un plot, puesto que sólo tenemos un margen de Ø a 31 de columna, y de Ø a 23 de fila. De todas formas habéis visto ya cual es la forma de tratar la pantalla, en el caso de que quisieramos la coordenación por pixel, tendríamos que hacer unos arreglos, ya que ahora los valores que pueden tomar B y C son de Ø a 191, y de Ø a 255 respectivamente.

De cómo hacer esto y otros igualmente interesantes respecto de la coordenación por pixels, requisito indispensable para conseguir unos efectos de animación idénticos a los de los juegos comerciales, hablaremos en el próximo número.



Dibujo de la estructura de la pantalla.



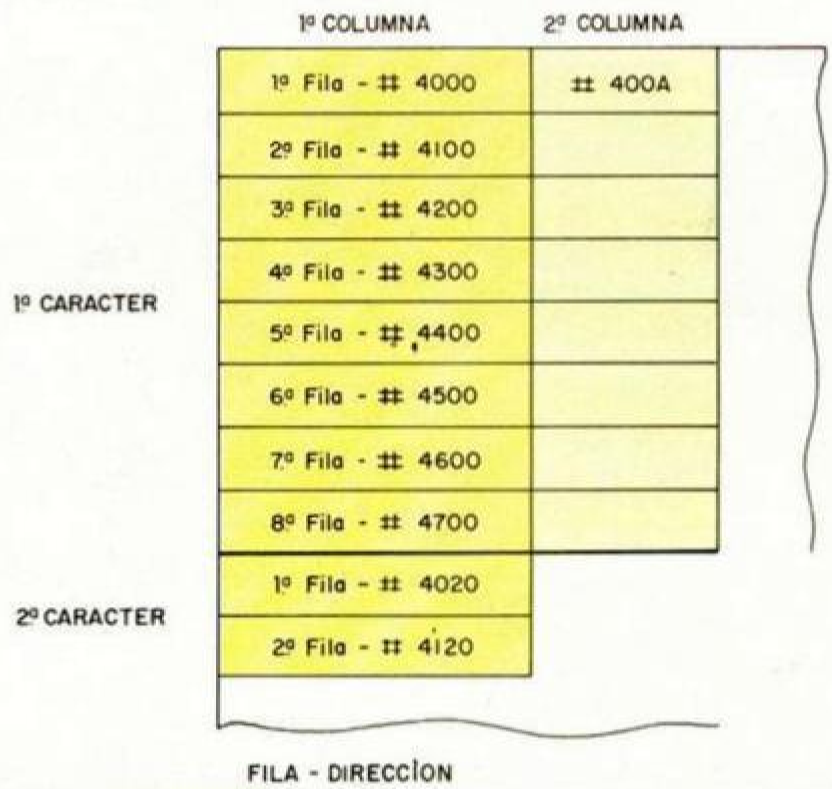
Estructura de un tercio de la pantalla.

DETALLE DE LA ESTRUCTURA DE LAS PRIMERAS LINEAS DE LA PANTALLA

La estructura de la memoria de pantalla es bastante compleja y nada mejor que esta rutina para manejarla desde Código Máquina.

En la figura se aprecia cómo el orden en que están dispuestas las direcciones de memoria, completa primero el 1.er byte de cada fila (en un sólo tercio de pantalla).

Luego completa el 2.º byte de cada fila y así hasta terminar con el primer tercio. Luego pasa al segundo y así hasta el final.



¡GUERRA al I.V.A.!

Suscríbete ahora a MICROHOBBY SEMANAL y benefíciate de un

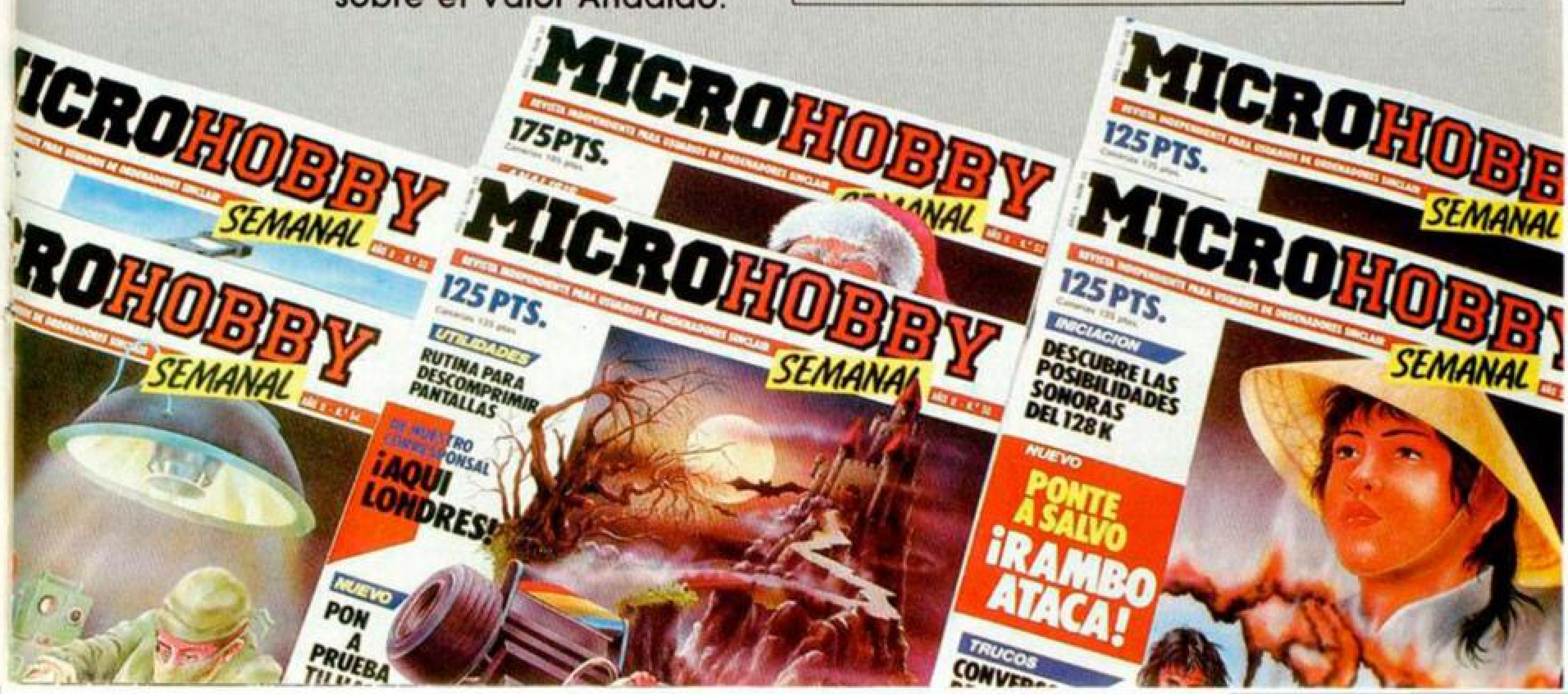
22% DESCUENTO

(Oferta válida sólo hasta el 28 de febrero de 1986)

MICROHOBBY SEMANAL **te ofrece ahora una oportunidad excepcional. Hasta el 28 de febrero de 1986 podrás suscribirte a nuestra revista sin tener que pagar el recargo correspondiente al Impuesto sobre el Valor Añadido.**

MICROHOBBY SEMANAL **lo abona por ti. Ahora puedes recibir** MICROHOBBY SEMANAL **en tu domicilio durante todo un año por sólo 5.250 ptas., es decir, 1.500 ptas. menos de su valor real.**

¡APROVECHA ESTA OPORTUNIDAD!



GENERADOR DE ROTULOS (y III)

Miguel SEPULVEDA y Adolfo PEREZ

En este último capítulo de los dedicados al generador de rótulos, explicamos el funcionamiento del programa a la hora de mostrar los displays de textos, así como los tipos de presentación y rutinas de Código Máquina utilizadas.

Mediante las opciones 1 y 2 del Menú principal, habíamos creado unos rótulos o textos que con la opción Salvar, se habían comprimido y almacenado en memoria, unos a continuación de otros, a partir de la dirección 32ØØØ. Todo lo creado hasta ahora, ha de ser preparado para poder mostrarlo por pantalla, según modalidades de presentación que dependen de unas claves que veremos a continuación.

OPCION 3: PREPARAR SALIDA POR PANTALLA

Esta opción se refiere a la configuración de datos para las distintas maneras que tenemos de presentar las pantallas en la TV. Lo primero que aparece en pantalla cuando elegimos la opción 3, es un cuadro que contiene todos los datos de los textos y rótulos creados hasta ese momento (vease la figura 4).

OR	TP	D.COM	C.FIN	N.BY	A	TM	C	TC
1	RO	32000	32606	606		0		0
2	RO	32606	32785	179		0		0
3	RO	32785	33267	482		0		0
4	TE	33267	33320	53		0		0

V=Ver. H=Hor. C=Cir. O=Lh. E=Lv.

Fig. 4. Ejemplo de presentación al seleccionar la opción 3.

Vamos a ver qué significa cada campo de esta figura.

- OR: Número de orden. Se refiere al orden con que hemos introducido los rótulos o textos.
- TP: Tipo (RO=Rótulos TE=Texto)
- D.COM: Dirección de comienzo de los datos comprimidos.
- D.FIN: Dirección donde terminan los datos.
- N.BY: Número de bytes que ocupa la pantalla de los rótulos o el texto en cuestión.
- A: Modo o código de apertura. Veremos más adelante qué significa.
- TM: Tiempo de exposición. Es el tiempo en segundos que permanece la pantalla del TV con la información deseada.
- C: Modo o código de cierre. Ya veremos qué significa.
- TC: Tiempo de cierre. Es el tiempo en segundos que permanece la pantalla del TV en negro antes de pasar a presentar el siguiente rótulo o texto.

En la parte inferior de la pantalla, aparecen una siglas que se refieren a los códigos de apertura y cierre al presentar los rótulos en la pantalla del TV, y su significado es el siguiente:

CODIGOS DE APERTURA Y CIERRE

- V: Cierre o apertura en vertical.
- H: Cierre o apertura en horizontal.
- C: Cierre o apertura en circular.
- O: Cierre o apertura en líneas horizontales.
- E: Cierre o apertura en líneas verticales.
- D: Desplazamiento de la pantalla hacia arriba. Sólo en cierres.

El ejemplo propuesto consta de tres rótulos y un texto, cuyas direcciones de comienzo y fin, así como su longitud, podremos observar en la figura. Los datos correspondientes a códigos de apertura y cierre están sin rellenar, y los tiempos de presentación y cierre están a cero. Mediante la opción 3, rellenaremos estos campos. Sigamos el programa para ver cómo se rellenarían.



F. L. FRONTAN

Accedemos a una u otra línea de información por su número de orden, por tanto, éste será el primer dato que tendremos que introducir. A continuación el programa nos pide los códigos de apertura y cierre para el rótulo o texto en cuestión.

Posibilidades de presentación

HORIZONTAL

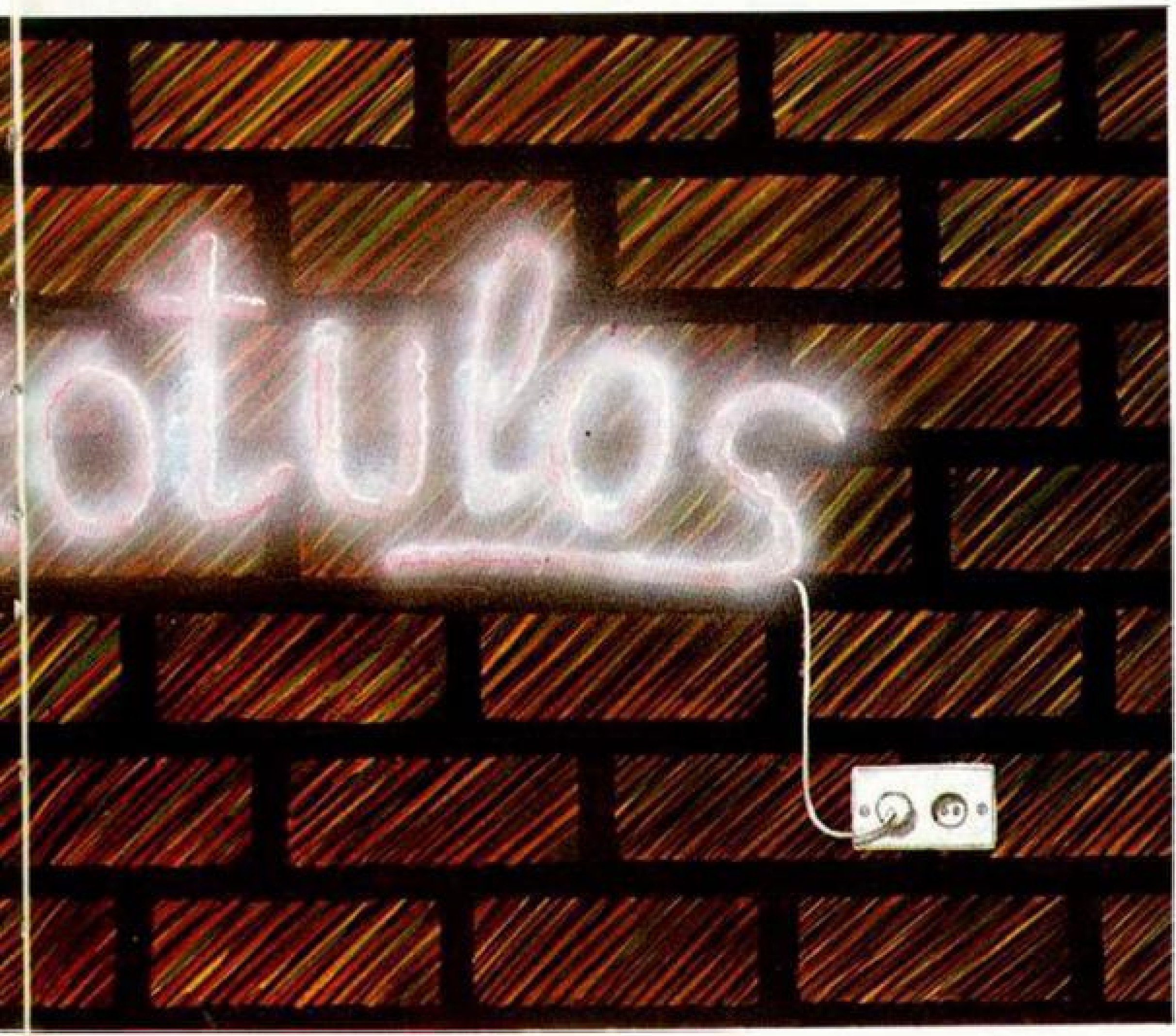
El rótulo se presenta en pantalla, y unas grandes rayas verticales aparecen de arriba a abajo y viceversa, hasta confluir en el centro. La apertura sería al contrario, es decir, partirían del centro las rayas hasta llegar a las partes alta y baja de la pantalla.

VERTICAL

Las rayas van o vienen, desde el centro hasta los laterales de la pantalla.

CIRCULAR

El efecto conseguido, es como si partiésemos de un punto en blanco que se va haciendo más grande cada vez hasta llenar la pantalla. Si es para cerrar, sería al contrario, a partir de la pantalla con el rótulo, se iría ennegreciendo por los cuatro lados hasta llegar al centro.



LINEAS VERTICALES Y HORIZONTALES

En los códigos de líneas verticales (E) o líneas horizontales (O), se presentan unas rayas que se entremezclan hasta hacer aparecer o desaparecer la pantalla según sea código de apertura o cierre. Dependiendo de que sea E u O, estas líneas serán verticales u horizontales.

Los códigos pueden estar mezclados entre sí, es decir, podremos utilizar un código para apertura y otro distinto para cierre.

Para un número de orden en particular, una vez dados los códigos, hemos de especificar cuánto tiempo queremos que el rótulo permanezca en pantalla y cuánto ha de transcurrir hasta que aparezca el siguiente.

Este proceso se repetirá para cada uno de los rótulos que hayamos definido.

Para los textos, solamente hemos de especificar los tiempos de presentación y cierre, puesto que siempre aparecen por la parte inferior de la pantalla.

Para salir de esta opción al menú principal, basta dar un número de orden que no exista.

Conforme vamos introduciendo los datos, el cuadro de la figura 4 se va modificando. Después de la definición de parámetros para los rótulos y textos de la figura 4, ésta queda modificada tal como aparece en la figura 5.

OR	TP	D.COM	C.FIN	N.BY	A	TM	C	TC
1	RO	32000	32606	606	h	1	v	1
2	RO	32606	32785	179	e	2	o	2
3	RO	32785	33267	482	c	3	d	1
4	TE	33267	33320	53	t	0	t	2

V=Ver. H=Hor. C=Cir. O=Lh. E=Lv

Fig. 5. Ejemplo de pantalla tras introducir los parámetros de tiempo (TM) y código (C).

OPCION 4: EJECUCION DE LO PREPARADO

Esta opción presenta en pantalla todo lo definido hasta ese momento, una vez haya sido preparado para salir por pantalla mediante la opción 3 del menú. La presentación se va haciendo por orden consecutivo, y tal como fueron definidos. Cuando todos los rótulos y textos hayan sido presentados nos aparecerá el menú principal.

OPCION 5: LIMPIAR MEMORIA

Con esta opción todo lo almacenado hasta ese momento es borrado dejando los punteros en las posiciones iniciales.

OPCION 6: INFORMACION DE LO CREADO

En cualquier momento podremos obtener información de lo creado utilizando esta opción. Nos aparecerá un cuadro similar al de las figuras 4 ó 5, con excepción de las claves de la parte inferior que no aparecen aquí, ya que no podremos modificar nada. Con cualquier tecla volveremos al menú.

OPCION 7: SALIDA DEL PROGRAMA

Esta opción detiene la ejecución del programa en la línea 810. Si queremos volver a ejecutar el programa sin necesidad de cargarlo de nuevo, basta hacer un GOTO 10.

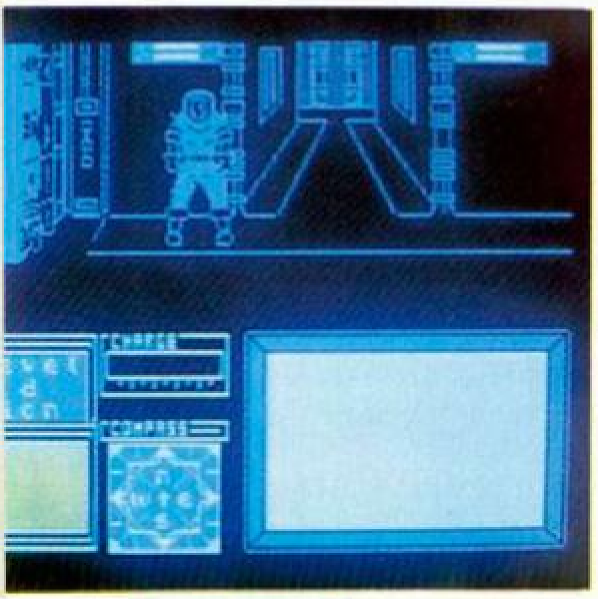
Para terminar el artículo vamos a enumerar las rutinas en código máquina utilizadas, y la dirección en que se encuentran.

RUTINAS DE CODIGO MAQUINA UTILIZADAS Y SUS DIRECCIONES

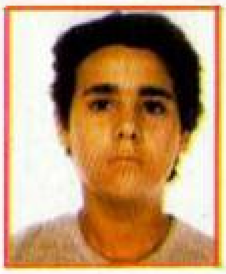
DIRECCION	NOMBRE	COMENTARIO
63392	CRHOR	Cierra pantalla en horizontal.
63416	CRLVE	Cierra pantalla en líneas verticales.
63442	ABHOR	Abre pantalla en horizontal.
63487	ABLVE	Abre pantalla en líneas verticales.
63538	CRVER	Cierra pantalla en vertical.
63552	CRLHO	Cierra pantalla en líneas horizontales.
63599	ABVER	Abre pantalla en vertical.
63626	ABLHO	Abre pantalla en líneas horizontales.
63704	CRCIR	Cierra pantalla en circular.
63754	ABCIR	Abre pantalla en circular.
63830	COMPP	Comprime la pantalla.
63970	ESPAP	Expande la pantalla.
64027	DESTX	Desplaza texto por la pantalla.
64121	DESPC	Hace scroll hacia arriba de un sacn.
64195	LMAYU	Muestra letras gigantes en pantalla.

LOS JUSTICIEROS DEL SOFTWARE

MARSPORT







«Un juego original»

• **Positivo:** Los gráficos son buenos pero sin llegar a ser perfectos. El movimiento de personajes sí que es una maravilla, así como el scroll de la pantalla.

La idea es buena pero parece ser muy complicada y enredosa.

• **Negativo:** El movimiento de las teclas es muy complicado. El sonido prácticamente no existe.

El tratamiento del color no es muy bueno ya que no distingue bien el color negro del personaje con el fondo rojo.

Resumiendo: Es un juego original. Impresiona por el movimiento del personaje que es muy natural.

Puntuación: 8

JAVIER VALERO. 15 años. Estudiante. Elche/Alicante.



«Graficos excepcionales»

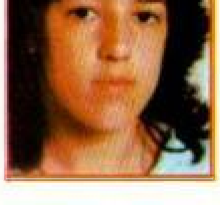
• **Positivo:** Los gráficos son excepcionales. Los reflejos son bastante buenos. Es original a tope aunque se parece a Dun Darach.

• **Negativo:** El sonido es bajo. El movimiento es lento.

Resumiendo: Lo mejor del juego, los gráficos. Es un programa divertido.

Puntuación: 7

CELIA MENENDEZ. 17 años. Estudiante. Madrid.



«Semejante al Dun Darach»

• **Positivo:** Los gráficos son bastante buenos. El movimiento es quizás lo mejor del juego. A medida que se adentra en él va adquiriendo un mayor interés, debido a la variedad de fases o niveles en las que vas encontrando muchas cosas nuevas que hacen más ameno el juego.

• **Negativo:** Es una copia idéntica, en cuanto a gráficos y movimientos se refiere, a Tir Nao Nog y Dun Darach, a simple vista es una continuación de ellos.

Resumiendo: Buena, pero poco original.

Puntuación: 6

PURIFICACION HERNANDEZ. 15 años. Estudiante. Burgos.



«Ausencia de sonido»

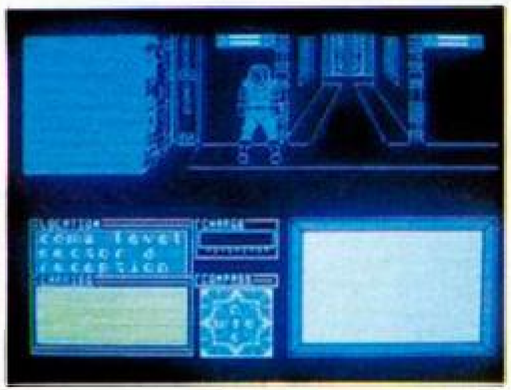
• **Positivo:** Tiene unos gráficos bastante buenos, especialmente el personaje central. Las pantallas dan una buena sensación de perspectiva y todos los objetos que aparecen en ella están bien logrados. El movimiento de nuestro personaje es bueno y bastante simpática su forma de andar. Buena capacidad de respuesta de las teclas.

• **Negativo:** La pantalla de presentación es bastante pobre. Se echa de menos la posibilidad de manejar el joystick. El sonido es prácticamente nulo. Cuesta bastante orientarse por los distintos pasillos, debido a que son similares entre sí.

Resumiendo: Un juego original con un alto nivel de dificultad y adictivo.

Puntuación: 7

JAVIER REDONDO. 18 años. Estudiante. Madrid.





«Entraña gran dificultad»

• **Positivo:** Los gráficos y el movimiento son bastate buenos. Es un juego entretenido y con mucha adicción así como dificultad.

• **Negativo:** El sonido no es muy bueno y si no se tiene dominio con los movimientos y cambios de perspectiva te puedes perder y pasarte mucho tiempo dando vueltas sin encontrar nada.

Resumiendo: Es un buen juego, pero difícil de acabar.

Puntuación: 7

JOSE CARLOS NUÑEZ. 18 años. Estudiante. Santander.



«Personaje: movimiento excelentes»

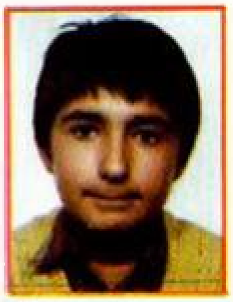
• **Positivo:** Los gráficos son muy buenos. Los movimientos son excelentes sobre todo el andar del personaje.

• **Negativo:** Aunque sea más o menos original tiene una gran parecido con los anteriores juegos de la compañía. La dificultad es muy grande.

Resumiendo: No es un juego muy divertido.

Putuación: 7

DANIEL GONZALEZ. 12 años. Estudiante. Madrid.



«Resulta algo monótono»

• **Positivo:** El movimiento del personaje central es magnífico y los gráficos son bastante aceptables. El scroll lateral está bastante conseguido.

• **Negativo:** Carece de sonido y es lioso, ya que *sólo nos guiamos por una* brújula. Aunque su dificultad es elevada no crea mucha adicción.

Carece de alicientes, caídas, golpes, etc., por lo que resulta muy monótono.

Resumiendo: Un programa regular.

Puntuación: 4

GORKA POLITE VILLAMAYOR. 12 años. Estudiante. Pamplona.



«Garantiza horas de entretenimiento»

• **Positivo:** Continuación de la saga de Tir Nao Nog, Dun Darach..., en una nueva aventura espacial. Los movimientos, como ya es usual, son excelentes.

En cuanto al escenario, está visiblemente enriquecido con ascensores, pantallas de videotexto, alienígenas amenazadores y un sin fin de sofisticaciones.

Los gráficos están muy logrados e interesantes.

• **Negativo:** Se hace un poco complicado orientarse en el plano por la gran similitud entre pantallas.

Resumiendo: Buen juego en general que garantiza horas de entretenimiento.

Puntuación: 7

MARIA A. GAGO. 18 años. Estudiante. Cádiz.

MICRO Manía

Sólo para adictos

CEDIDO POR MICRO Manía

ROBIN OF SHERWOOD

En el número 55 de nuestra revista publicábamos un pequeño guión gracias al cual podíais escapar con facilidad de la mazmorra en la que aparecíamos prisioneros al principio de esta emocionante aventura. Ahora, os mostramos una guía más completa que os ayudará en muchas más situaciones embarazosas. Para utilizar correctamente estas ayudas tan sólo deberéis reemplazar los números que aparecen como respuestas por la palabra correspondiente del diccionario. Pero cuidado, estas soluciones no tienen por qué ser tecleadas al pie de la letra, sino que sólo sirven como ligeras indicaciones de qué es lo que debéis de hacer.

— No puedes salir de la mazmorra.
1 3 10 12 2 7 5 4 13 16 18 8 11 19 20 23 24 21 8 6

— No puedes encontrar tus armas.
9 25 29 27

— No puedes encontrar tu banda.
26 30 34 22 8 17 19 31 14 33 28

— Si tienes problemas para rescatar a Marion.
35 8 37 31 36 8 38 19 15 13 39 33 40

— No puedes encontrar oro para pagar a la monja.
16 41 8 42 43 45 19 8 44

— No puedes abrir la caja.
48 46 29 47 32 8 49

— No puedes encontrar al ladrón.
16 41 29 50 54 32 51 41 29 52 32 8 53

— No consigues satisfacer a los templarios.
16 41 8 42

— No puedes entrar en el castillo.
55 8 59

— No logras encontrar una Touchstone.
16 60 8 56 58

— Sigues perdiendo tu oro.
61 8 49 62

No consigues encontrar a Belleme.
16 41 29 57 2 29 63 66 64 29 65

— No consigues encontrar a Belleme.
16 41 29 57 3 29 63 66 64 29 65 68 29 69 32 29 67

— No puedes encontrar todas las Touchstones
6 41 8 42

— Si aún no consigues encontrar todas las Touchstones.
70 8 75 73

— Si a pesar de todo no consigues encontrar las Touchstones.
71 76 29 74 48 78 41 29 77 72 19 29 40 31 81 80 79

Dictionary

1. climb
2. grab
3. on
4. strangle
5. ankle
6. bolt
7. an
8. a
9. go
10. somebody's
11. sword
12. shoulders
13. someone
14. find
15. kill
16. look
17. fight
18. for
19. and
20. use
21. undo
22. after
23. it
24. to
25. into
26. collect
27. waterfall
28. camp
29. the
30. Little
31. then
32. of
33. your
34. John
35. win
36. take
37. competition
38. hostage
39. with
40. prize
41. in
42. box
43. some
44. statue
45. sacks
46. need
47. services
48. you
49. thief
50. bushes
51. Leaford
52. presence
53. lady
54. south
55. hitch
56. special
57. castle
58. tree
59. ride
60. up
61. there's
62. about
63. sand
64. through
65. door
66. flats
67. corridor
68. at
69. end
70. search
71. drop
72. circle
73. satanist
74. Touchstone
75. dead
76. all
77. stone
78. have
79. happens
80. what
81. see

MICRO HITS

1. **EXPLODING FIST** *(Melborne House)*
2. **NIGHTSAHADE** *(Ultimate)*
3. **SPY HUNTER** *(U. S. Gold)*
4. **BRUCE LEE** *(U.S. Gold)*
5. **NIGHT LORE** *(Ultimate)*
6. **HYPERSPORT** *(Imagine)*
7. **BACK TO SKOOL** *(Mikrosphere)*
8. **MATCH DAY** *(Ocean)*
9. **UNDERWURLDE** *(Ultimate)*
10. **DECATHLON** *(Ocean)*
11. **WORLD SERIES BASKETBALL** *(Imagine)*
12. **HERBER'ST DUMMY RUN** *(Mikro-Gen)*
13. **FRANK BRUNO'S BOXING** *(Elite)*
14. **HIGHWAY ENCOUNTER** *(Vortex)*
15. **DUN DARACH** *(Gargoyle Games)*
16. **SPY VS SPY** *(Beyond)*
17. **WORLD SERIES BASEBALL** *(Imagine)*
18. **NODES OF YESOD** *(Odin)*
19. **FIGHTING WARRIOR** *(Melbourne House)*
20. **GYROSCOPE** *(Melbourne House)*

CONSULTORIO

Falta de memoria

Me he comprado un Spectrum Plus hace poco, y al teclear el programa QUASIMODO del número 6 de MICROHOBBY, me ocurre que cuando llego a la línea 7100, al oprimir cualquier tecla, suena un pitido y no obedece el ordenador ninguna orden. Una cosa parecida me pasa con Andrómeda en la línea 5000. ¿A qué se debe ésto?

Gabriel LOPEZ - Madrid

☐ Lo que usted nos indica es síntoma de falta de memoria. Dado que se trata de un PLUS, debería tener 48K de RAM, pero es posible que tenga parte de memoria averiada y esté trabajando con menos.

Teclee: PRINT PEEK 23732+256*PEEK 23733 y, si la respuesta es un número distinto de 65535, lleve su ordenador a reparar (esperamos que aún esté en garantía).

«MOVE»

En el programa PIPELINE del número 2, aparece «MOVE»:

1180 LET B(2,2)=1: LET MOVE=1

1190 LET MOVE=MOVE+1

Y en más, pero no sube la línea en mi Spectrum Plus, apareciendo el signo parpadeante de error. ¿Por qué?, ¿se puede cambiar por otra sentencia o comando?

También quisiera saber si el signo "^" (acento circunflejo) que aparece en algunos programas, se puede crear como gráfico o cambiarlo por otro carácter.

Félix NOVILLO - Madrid

☐ Esta duda ya le surgió a otro lector. En el programa que nos indica, la palabra «MOVE» debe teclearla como una variable (letra a letra), no como un comando.

El signo "^" se utiliza en las impresoras en lugar de la «flecha hacia arriba» (código 94) se trata, por tanto, de signos equivalentes.

«SPRITES»

¿Cómo podría hacer que un grupo de «pixels» se mueva de izquierda a derecha de la pantalla, sin utilizar el engorroso sistema de hacerlo «de gráfico en gráfico»?

Según tengo entendido, la solución son los «sprites», pero no sé ni cómo son ni cómo se crean.

Alejandro FERNANDEZ - Asturias

☐ Efectivamente, la solución (en Basic) son los «sprites», pero el Basic del Spectrum no permite crear «sprites», así que no queda más remedio que recurrir al código máquina.

En nuestro artículo «Gráficos en movimiento» de los números 16, 17, 18 y 19, hablábamos de ello, y seguiremos hablando en sucesivos artículos, así como en el Curso de Código Máquina.

Código Máquina

¿Utilizan todos los ordenadores el mismo código máquina? y si es así, ¿un programa escrito en código máquina de un ordenador, no serviría a otro de distinta marca?

¿No hay ningún programa que pase del Basic del Spectrum al del Amstrad?

ANTONIO - Valencia

☐ Utilizan el mismo código máquina todos los ordenadores que tengan el mismo microprocesador (tienen un Z-80 el Spectrum, Amstrad y todos los MSX), pero esto no quiere decir que los programas sean compatibles ya que, en un programa en código máquina, influye decisivamente la arquitectura del ordenador.

No tenemos noticias de que exista un programa que traduzca el Basic del Spectrum al del Amstrad; pero, en muchos casos, la traducción sería imposible de realizar por un programa, ya que tanto los comandos como la forma de trabajar son distintos. Si se desea efectuar la traducción, no hay más remedio que hacerlo «a mano», y le aseguramos que no es tarea fácil.

Spectrum 128K

Estas navidades tengo pensado comprarme un Spectrum 128K, pero desearía que me aclararan lo siguiente: ¿es cierto que dicho ordenador no es 100% compatible con el software del Spectrum Plus? ¿Cuando no esta en modo Spectrum qué lenguaje utiliza? ¿Se pueden utilizar programas del QL en modo 128K?

Adolfo MARTIN - Madrid

☐ Según parece, el fabricante asegura que el Spectrum 128K es totalmente compatible con el software existente para su hermano menor, no obstante, será el tiempo quien saque a relucir las posibles incompatibilidades.

En modo 128K, utiliza también el lenguaje Basic, pero con una serie de posibilidades que le hacen más cómodo de utilizar.

En ningún caso es compatible con el QL ni con su software.

Grabador de EPROM

En relación con el grabador de EPROM publicado ul-

timamente en su revista, desearía me dijeran si es posible almacenar en la memoria principal parte de la EPROM en vez de todo su contenido. Si lo es, cómo hacerlo, teniendo en cuenta que la rutina que lo haga ha de ser llamada desde un programa. Así mismo, ¿qué tiempo medio tarda en cargarse 1K?

A. BERNAT - Tarragona

☐ Para copiar en RAM un trozo de la EPROM, puede utilizar la opción «COPIAR EN RAM», indicando la dirección inicial de EPROM, la dirección final de EPROM y la dirección inicial de RAM; las direcciones de EPROM debe darlas partiendo desde 0000, ya que el grabador genera sus propias direcciones.

Si lo que desea en, como suponemos, cargar subrutinas desde EPROM para utilizarlas en sus programas (una especie de «disco EPROM»), será mejor que diseñe su propia rutina en código máquina que lo haga a través de los ports; en los artículos correspondientes se indica cuales son los ports que ha de usar en cada caso. Si usara para ésto el programa que nosotros diseñamos para controlar el grabador, estaria ocupando inecesariamente una cantidad mayor de memoria.

Respecto a su otra pregunta, 1K tarda en grabarse en EPROM aproximadamente 61 segundos. Sin embargo, la carga en RAM es prácticamente instantánea.

«GENS 3» y «MONS 3»

Si yo quiero modificar un programa en C/M que tengo en memoria (lease una subrutina de la ROM o una de las que publicáis en vuestra revista), primero tengo que ver los nemónicos y, para ello, utilizar un desensamblador. Pero, a la hora de modificar, me gustaría tener un desensamblador-ensamblador que me muestre el desensambladro en pantalla y pueda modificarlo como si se tratara de un ensamblador.

¿Existe algún desensamblador que haga ésto?

Eduardo MARTIN - Valladolid

☐ Lo que usted pide es algo realmente difícil; afortunadamente, tiene la suerte de poseer uno de los pocos ordenadores donde podrá hacerlo; sólo le falta el paquete de aplicación de HISOFT compuesto por los programas «GENS 3» y «MONS 3», de lo mejor que se ha escrito para Spectrum.

El «GENS 3» ya es conocido por muchos de nuestros lectores como el mejor ensamblador que existe para el Spectrum. El «MONS 3» no tiene nada que envidiar a su «hermano»; se trata de un magnífico desensamblador-monitor que permite una enorme gama de posibilidades, entre ellas, la de desensamblar un bloque de memoria y enviar el código fuente al «GENS 3» para poder trabajar con él.

A este respecto, le recomendamos la lectura del artículo aparecido en la página 70 de nuestro número ESPECIAL, donde se comentan ampliamente estos dos programas.

Basic avanzado

Estoy haciendo un programa que necesita muchas sentencias del tipo IF ... THEN GO TO... seguidas (IF r=1 THEN GO TO 100: IF r=2 THEN GO TO 200, etc) ¿Cómo puedo disminuir el número de estas sentencias?

Ese mismo programa necesita que, según una fórmula almacenada en memoria, (por ejemplo: a=3•r•4/t), acepte cualquier dato y devuelva el resultado. ¿Cómo puedo hacer ésto?

El cargador universal de código máquina al que suele hacer referencia la revista MICROMANIA en su sección de cargadores, ¿es el mismo que el aparecido en vuestra revista?

Iñaqui GRAO - Bilbao

☐ La forma típica de hacer lo que usted pide seria:
ON r GO TO 100,200,...

Por desgracia, el Spectrum no admite la instrucción ON...GO TO, pero es posible simularla de la siguiente forma:

GO TO 100•r

Eso sí, deberá tener sumo cuidado con los valores de «r».

Respecto a su segunda pregunta, es posible hacerlo en el Spectrum (y es uno de los pocos ordenadores cuyo Basic lo permite). Pruebe el siguiente programa:

```
10 INPUT "Fórmula a aplicar ?";a$
20 INPUT "Dato de entrada ?";a
30 LET b=VAL a$
40 PRINT "Resultado=";b
```

La línea 10 pide que se le introduzca la fórmula a aplicar que, lógicamente, deberá estar en función de «a» que será la variable de entrada. La línea 30, simplemente, calcula la fórmula aplicando sobre "a$" la función "VAL". Pruebe a introducir primero "a•2—7/a" como fórmula, y luego «7» como valor de «a»; deberá obtener «13» como resultado. Tenga cuidado al introducir las fórmulas, o le esperarán muchos "Nonsense in Basic".

En cuanto al cargador de código máquina, sí, es el mismo.

Hexadecimal

Soy un asiduo lector de MICROHOBBY SEMANAL. No tengo ordenador, aunque espero conseguirlo estas navidades, y puede ser que sea ésta la razón de que no entienda la forma de utilizar la cara posterior de la tabla «Conversión HEX-DEC» de la MICROFICHA T-1.

Juan LUPIAÑEZ - Barcelona

☐ La tabla que nos indica sirve para convertir números de «Decimal» (base 10) a «Hexadecimal» (base 16). El sistema de numeración Hexadecimal se usa para programar en código máquina, por lo que la ficha sólo le será útil si piensa programar en este lenguaje.

La cara posterior de la misma le da el valor del octeto superior en números de 16 bits. La forma de usarla se ve mejor con un ejemplo: Supongamos que queremos pasar a decimal el número 5D4Fh; primero miramos 5D en la cara posterior y obtenemos 23808; luego buscmos el valor 4F en la cara anterior y obtenemos 79; sumamos 23808 + 79 = 23887 que es el valor de 5D4Fh en Decimal.

AVISO A NUESTROS LECTORES

Os sugerimos que la correspondencia que nos enviéis para las diferentes secciones de nuestra revista, lo sea en distintos sobres y explicando claramente en el exterior el contenido del mismo («CONSULTORIO», «OCASION», «TRUCOS», etc.), ya que el camino que siguen las cartas dirigidas a cada sección es completamente distinto. De esta forma nos ayudaréis a evitar innecesarios retrasos. Gracias por vuestra colaboración.

DE OCASION

● VENDO Spectrum 48 K, con sus cables, cinta horizontes, y libros de instrucciones. Regalo varias revistas. Comprado hace 10 meses. Precio 30.000 ptas. También vendo Lápiz Optico DK'Tronics, con su cinta e intrucciones por 5.000 ptas. comprado hace 10 días. Preguntar por José al Tel. (965)856480. Noches.

● CLUB de soft Snoppy más que un club un amigo. Interesados escribir a José Miguel Gómez García. Avda. Andalucía, 7, 8.º B. Málaga. Tel. (952)344330.

● VENDO Spectrum 48 K, totalmente nuevo, con accesorios, manuales en español varias revistas y libros de programación. Todo ello por sólo 23.000 ptas. Llamar por las tardes de 7 a 10 a Mario. Tel. 4041285 de Madrid.

● VENDO Interface (programable) Indescomp, a estrenar, en perfecto estado y con garantía. Adquirido en septiembre. Interesados escribir a Miguel. Cañaveral, 84. Dup. Madrid. Tel. 2155172.

● VENDO Spectrum 48 K, Microdrive, Interface 1, más de 50 revistas, garantía en blanco, manuales en castellano y el libro Basic Sinclair de Microhobby. Todo por el precio de 59.000 ptas. Interesados llamar al Tel. 7770167 de Madrid.

● VENDO Joystick Quick Shot I e Interface Kempston. Ambos periféricos por sólo 2.500 ptas. y por separado 1.500 cada uno. Interesados contactar con Pedro Vera de la Rocha. Rbla. S. Esteban, 11, Atco. Ripollet (Barcelona). Tel. 6922253.

● COMPRO cinta original Horizontes por 3.000 ptas. Llamar al siguiente Tel. (94)670255 de Vizcaya. Preguntar por Iñigo.

● VENDO fotocopias de libros para Spectrum: «The complete Spectrum Rom Disassembley», «Iniciación al Código Máquina», «Basic Básico», etc. Llamar al Tel. (91)2163685. También desearía que me facilitaran libros, no importa fotocopias, de lenguaje de programación, exceptuando Basic. Dirección: Javier Martín Rueda. Nueva Zelanda, 52, 3.º. 28025 Madrid.

● SE VENDE consola de videojuegos Home video Game (SD-05), con dos joystick incorporados. Interesados dirigirse al Tel. (91)7347419 (9 a 10 noche).

● ALGORTA interesados en formar un club de intercambio de ideas (no de software) contactar con Matías. Tel. 4695923 (fines de semana). También vendo Spectrum Plus, barato y nuevo, completo con garantía, hasta agosto del 86, escribir a Matías Vaquero. Piñabeko, 1, 1.º derecha. Algorta (Vizcaya).

● VENDO Spectrum Plus, manuales, cables, alimentación, amplificador de sonido, interface joystick, salida monitor, computador Save/Load, revistas, etc. Contactar con Carlos Rodríguez. Carreño Miranda, 5, 3.º Avilés (Asturias). Tel. (985)564923.

● VENDO Eproms, Micros y gran variedad de circuitos integrados, etc. Mando lista totalmente gratis. Interesados dirigirse a Miguel Angel Clavijo. José Antonio, 5, 1.º izquierda. La Cuesta (Sta. Cruz de Tenerife).

● HAGO protecciones en Turbo, con Basic muy protegido y Código Máquina también Turbo sin rayas, para más información llamar al Tel. (93)3479835.

● VENDO ZX Spectrum 48K, teclado Indescomp Multifunción II por sólo 35.000 ptas. Interesados llamar al Tel. (976)273301. Preguntar por Alberto de Andrés Mata. Lapuyade, 11, 2.º D. Zaragoza.

● VENDO ordenador Spectrum 48 K, totalmente nuevo, con accesorios, manual en castellano, garantía, dos libros por sólo 30.000 ptas. Interesados contactar con José Antonio Suárez Márquez. Vázquez Varela, 19, 3.º B. Vigo (Pontevedra). Tel. (986)411895.

● VENDO Spectrum 16 K, completo con cables, fuente de alimentación, etc. Está en buen estado. Precio 18.000 ptas. Urge. Tel. (93)2030118. Preguntar por Javi. Barcelona.

● VENDO video-juegos Atari modelo CX 2600, más unidad de alimentación, cuatro mandos, 2 cartuchos y los cables por 24.000 ptas. Todo en buen estado. También lo cambiaría por Spectrum 48 K. Interesados escribir a Antonio Flores Sánchez. García Lesmas, 2, 2.º. Valladolid.

● VENDO Spectrum 48 K, como nuevo, con todos sus accesorios para su funcionamiento, manual de instrucción, por 28.000 (escudos). Escribir a Fco. José Almeida e Silva. Apartado, 21, 5051 Peso da Régua. Codex. Portugal.

● CAMBIO Joystick Quick Shot I, junto con 7 juegos por un Joystick Quick Shot II. Apd. 116 de Algorta. (Vizcaya).

● VENDO ordenador ZX Spectrum 48 K Ram, en perfecto estado, incluye Resert On-Off para el Spectrum. Precio 24.500 ptas. Interesados llamar al Tel. 6991669 ó bien escribir a Davide Montemayor. Ciudad Real, 40, 3.º D. Parla (Madrid).

● CAMBIARÍA ordenador Oric Atmos 48 K (44.700 bytes libres) mas 60 listados de programas, revistas, fuente de alimentación y conexiones de cassette y T. V., mas dos manuales en castellano y uno en inglés, libro de Código Máquina. Todo ésto por un ordenador Commodore 64 con conexiones, T. V. y cassette. Interesados dirigirse a Rafael Santiago Sánchez. Crta. de Motril, 2, 3.º C. Armilla (Granada).

● VENDO ZX Spectrum 48 K, con Interface de Joystick programable, Joystick Captain Grant y cassette Sanyo portátil con adaptador de corriente todo por el precio de 40.000 ptas. (Tiene garantía por 6 meses). También vendo video-juego Philips GT-7000 con 8 cartuchos de juegos, por 12.000 ptas. Llamar al Tel. 4604623 (mañanas). También pueden escribir a José Pérez Valiente. Eduardo Rivas, 17. Madrid.

● ESTOY interesados en la adquisición del Interface I y un Microdrive para Spectrum. Ofertas a Fernando Ramírez García. Monjas, 1, 2.º Ciudad Real. Tel. (926)226467.

● VENDO Spectrum 48K, con cables, fuente de alimentación y cinta de demostración por sólo 31.000 ptas. Regalo Interface joystick. Contactar con Michael Lepoutre. Camino de Hoyarrasa, 67. La Moraleja, Madrid. Tel. 6501046.

● VENDO ZX-81, con ampliación de memoria a 16K, completo con instrucciones en castellano y un juego de ajedrez. Por sólo 16.000 ptas. Dirección: Javier Ondarroa. Plaza Julio Lazurtegui, 16, 8.º Bilbao. Tel. 4350344.

● VENDO video-juego Atari, casi nuevo, con 4 cartuchos, y 4 mandos. Precio: 10.000 ptas. Llamar al Tel. 3232328 de Barcelona. Preguntar por Neus.

● VENDO Spectrum Plus, en perfecto estado, con sus respectivos cables, cassette de demostración, manual de instrucciones en español, interface II, joystick marcad Quick Shot II, por sólo 40.000 ptas. Interesados llamar al Tel. (92)648624.

● VENDO C-16 (Commodore 16) con cables, cassette y fuente de alimentación. Gran capacidad de sonido y con Basic fácil. Llamar el Tel. (985)380783. Precio sede 25.000 ptas. (negociables). Preguntar por Gil Andrés (1,30 a 2,30 h).

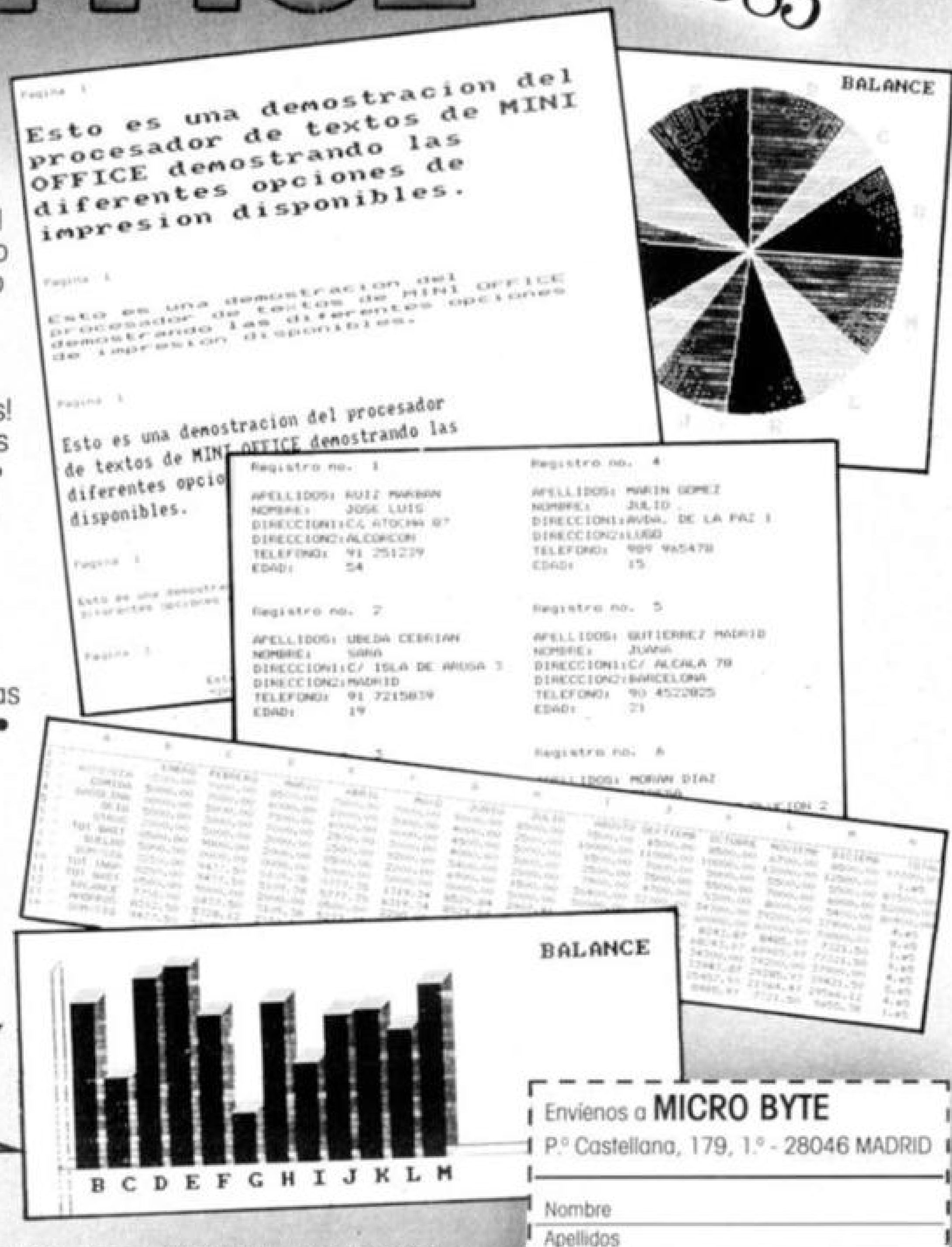
● CAMBIO consola de video-juegos SD-050C en perfecto estado, poco usado, por Joystick más Interface. Interesados escribir a Wenceslao Lumeras Amador. Braulio Navas, 37. Hervás (Cáceres), o bien llamar al Tel. (927)481218.

MINI OFFICE

PROGRAMA DEL AÑO EN INGLATERRA

FINALIST BRITISH MICRO COMPUTING AWARDS 1985

© DATABASE PUBLICATIONS



1 PROCESADOR DE TEXTOS

¡Ideal para escribir cartas e informes! Características: Visualización continua del tiempo • Contador de palabras (indicando las palabras por minuto) • Texto normal o doble, en pantalla o impresora.

2 HOJA DE CALCULO

¡Utiliza tu micro para controlar tus cuentas! Características: Cifras visualizadas en filas y columnas • Actualización permanente • Actualización reflejada instantáneamente en toda la hoja • Grabación de los resultados para futuras modificaciones.

3 GRAFICOS

¡Convierte esos números en maravillosos gráficos! Características: Gráficos de barras en tres dimensiones • Gráficos de pastel • Histogramas.

4 BASE DE DATOS

¡Igual que los archivos de la oficina! Características: Cargar ficheros con solo pulsar una tecla • Clasificación • Modificación • Listados • Búsqueda.

*En Castellano
*Servimos en 48 Horas

4 PROGRAMAS AL PRECIO DE 1
P.V.P. 3.200* Pts

*Versión disco AMSTRAD P.V.P. **3.900 ptas.**







Envíenos a **MICRO BYTE**
P.º Castellana, 179, 1.º - 28046 MADRID

Nombre
Apellidos
Dirección
Población
D.P. Teléfono

Deseo que me envíen ☐ ejemplar/es del programa **MINI OFFICE**

PARA EL MICROORDENADOR SEÑALADO

☐ AMSTRAD ☐ COMMODORE ☐ SPECTRUM
☐ AMSTRAD VERSION DISCO
Sin gastos de envío

INCLUYO TALON NOMINATIVO ☐
CONTRA-REEMBOLSO ☐

Pedidos por teléfono
91 - 442 54 33/44


DINAMIC: ALGO MUY ESPECIAL PARA TU ORDENADOR
SPECTRUM
DINAMIC
SPECTRUM
OLE, TORO
DINAMIC
1.º EDICION AGOTADA
2.º A LA VENTA
SPECTRUM
CAMELOT
WARRIORS
NUEVO
YA A LA VENTA
SPECTRUM · AMSTRAD
ROCKY
SPECTRUM · AMSTRAD
ABU SIMBEL
PROFANATION
SPECTRUM · AMSTRAD
WEST BANK
DINAMIC
DINAMIC
Tiendas y distribuidores (91) 447 34 10
Pedidos contra reembolso (91) 715 00 67
DINAMIC SOFTWARE: EN LAS MEJORES TIENDAS DE INFORMATICA